

# NetScatter: Enabling Large-Scale Backscatter Networks

Mehrdad Hessar<sup>†</sup>, Ali Najafi<sup>†</sup> and Shyamnath Gollakota

*University of Washington*

<sup>†</sup>Co-primary Student Authors

**Abstract** – We present the first wireless protocol that scales to hundreds of concurrent transmissions from backscatter devices. Our key innovation is a distributed coding mechanism that works below the noise floor, operates on backscatter devices and can decode all the concurrent transmissions at the receiver using a single FFT operation. Our design addresses practical issues such as timing and frequency synchronization as well as the near-far problem. We deploy our design using a testbed of backscatter hardware and show that our protocol scales to concurrent transmissions from 256 devices using a bandwidth of only 500 kHz. Our results show throughput and latency improvements of 14–62x and 15–67x over existing approaches and 1–2 orders of magnitude higher transmission concurrency.

## 1 Introduction

The last few years have seen rapid innovations in low-power backscatter communication [20, 29, 18, 15, 17], culminating in long range and reliable backscatter systems [25, 22, 27]. These designs enable wireless devices to communicate at microwatts of power and operate reliably at long ranges to provide whole-home or warehouse coverage. To achieve this, they employ low-power coding techniques such as chirp spread spectrum, to decode weak backscatter signals below the noise floor [22, 25] and deliver long ranges.

While these long range backscatter systems are promising for enabling power harvesting devices (e.g., solar and vibrations) as well as cheap and small Internet-connected devices that operate on button-cells or flexible printed batteries, they primarily work at the link layer and are not designed to scale with the number of devices — all these prior designs [25, 22, 27] are evaluated in a network of 1–2 devices.

Our goal in this paper is to design a network protocol that enables these low-power backscatter networks to support hundreds to thousands of concurrent transmissions. This is challenging because the resulting design must operate reliably with weak backscatter signals that can be close to or below the noise floor. To this end, we present *NetScatter*, the

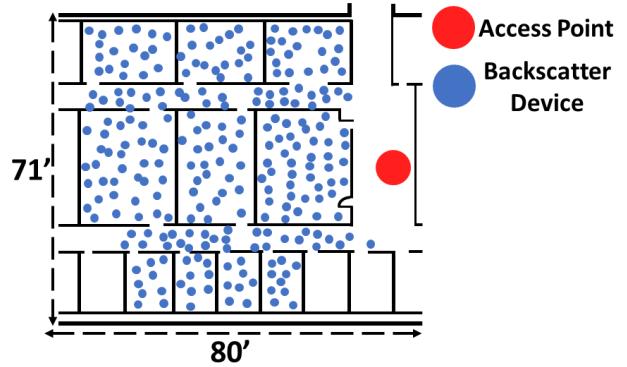


Figure 1: **Large-Scale Network Deployment of Backscatter Devices.** We deploy 256 backscatter devices across a floor of an office building covering multiple rooms.

first wireless protocol that can scale to hundreds and thousands of concurrent transmissions from backscatter devices. Our design enables concurrent transmissions from 256 devices over a bandwidth of 500 kHz. Consequently, it can support transmissions from a thousand concurrent backscatter devices using a total bandwidth of only 2 MHz.

Our key innovation is a distributed coding mechanism that satisfies four key constraints: i) it enables hundreds of devices to concurrently transmit on the same frequency band, ii) it can operate below the noise floor while achieving reasonable bitrates, iii) its coding operation can be performed by low-power backscatter devices, and iv) it can decode all the transmissions at the receiver using a single FFT operation, thus minimizing the receiver complexity.

We introduce *distributed chirp spread spectrum coding*, which uses a combination of chirp spread spectrum (CSS) modulation and ON-OFF keying. In existing CSS systems (e.g., LoRa backscatter [25]), the AP transmits a continuous wave signal which each device backscatters and encodes bits using different cyclic shifts of a chirp signal. In contrast, in our distributed CSS coding, we assign a different cyclic shift of the chirp to each of the concurrent devices. Each device then uses ON-OFF keying over these cyclic shifted chirps to convey bits, i.e., the presence and absence of the corresponding cyclic shifted chirp correspond to a ‘1’ and ‘0’.

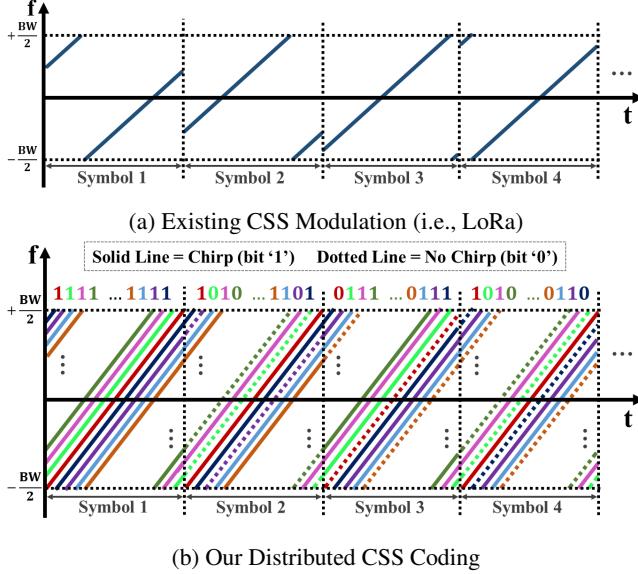


Figure 2: **NetScatter Overview.** In traditional CSS systems, a single device uses different cyclic shifts to convey bits. In distributed CSS coding, each cyclic shift is assigned to a different backscatter device. Each device then uses the presence and absence of cyclic shift to send ‘1’ and ‘0’ bits.

bit respectively, as shown in Fig. 2. Note that in comparison to existing CSS systems where each device transmits  $\log_2 N$  bits using  $N$  cyclic shifts, our distributed design enables  $N$  concurrent devices, each of which transmits a single bit using ON-OFF keying. Thus, our design transmits a total of  $N$  bits within a chirp duration, providing a theoretical gain of  $\frac{N}{\log_2 N}$ .

Our design leverages the fact that creating concurrent cyclic-shifted chirps at a single device requires distributing its transmit power amongst all the cyclic shifts, which reduces the ability of the receiver to decode each chirp. Instead we generate concurrent cyclic-shifted chirps across a distributed set of low-power devices in the network. This allows us to efficiently leverage the coding gain provided by chirp spread spectrum under the noise floor [9]. Further, we can decode all the concurrent transmissions using a single FFT operation, since cyclic shifting the chirps in the time domain translates to offsets in the frequency domain.

Using the above distributed coding mechanism in practice, however, is challenging for two key reasons.

- *Near-far problem.* A fundamental problem with enabling concurrent transmissions is that signals from a nearby backscatter device can overpower a farther concurrent device. To address this issue, we introduce two main techniques. First, we present a power-aware cyclic shift allocation technique in §3.2.3, where lower SNR devices use much different cyclic shifts than higher SNR devices. We show that such an allocation can allow backscatter devices that have an SNR difference of up to 35 dB to be concurrently decoded. Second, to account for channel variations over time, we develop a zero-overhead power adaptation al-

gorithm where backscatter devices use reciprocity to estimate their SNR at the AP, using the signal strength of the AP’s query message. The backscatter devices then adjust their transmission power to fall within the tolerable SNR difference. Since this calibration is done independently at each backscatter device using the AP’s query, it does not require additional communication overhead at the AP.

- *Timing synchronization.* The above design requires all the devices to start transmitting at the same time so as to enable concurrent decoding. However, hardware variation and propagation delays of different devices can make it challenging for hundreds of devices to be tightly synchronized in time. To avoid this coordination overhead, we leave gaps between cyclic shifts to ensure that concurrent devices are sufficiently distinguishable and can be decoded. We explore the trade-off between the required gaps and the chirp bandwidth in §3.2.1.

We implement *NetScatter* on a testbed of backscatter devices. We create backscatter hardware that implements *NetScatter* and includes circuits to perform automatic power adaptation before each transmission. We deploy our backscatter testbed with 256 devices in an office building spanning multiple rooms as shown in Fig. 1. We implement our receiver algorithm using USRP X-300 software-defined radios. Our results reveal that over a 256 node backscatter deployment, *NetScatter* achieves a 14–62x gain over prior long-range backscatter systems [25] for its end-to-end link layer data rates. The key benefit however is in the network latency which sees a reduction of 15–67x.

**Contributions.** Our paper demonstrates, to the best of our knowledge, the first network protocol that achieves orders of magnitude more concurrent transmissions than existing backscatter systems. The closest work to our design is *Choir* [12] *in the radio domain*, which decodes concurrent transmissions from 5–10 LoRa radios at a software radio. *Choir* leverages frequency imperfections to disambiguate between LoRa radios. However, backscatter devices achieve low power operations by running at a lower frequency (1–10 MHz) than radios (900 MHz) and thus have much smaller frequency differences between backscatter devices. This severely limits the ability to rely on frequency imperfections to disambiguate between a large number of backscatter devices (see §2.2). In contrast, our distributed chirp spread spectrum coding mechanism provides a systematic approach to enable large scale backscatter networks.

## 2 CSS Primer & Existing Approaches

### 2.1 Primer on Chirp Spread Spectrum

In CSS, data is modulated using linearly increasing frequency signals or upchirps. The receiver demodulates these symbols in a two step process. First, it de-spreads these up-

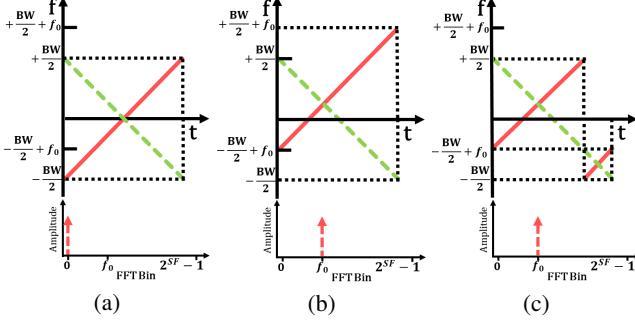


Figure 3: **CSS Primer.** We show upchirp and downchirp symbols and FFT results of their multiplication. (a) Baseline upchirp symbol, (b) frequency shifted upchirp symbol and (c) cyclically shifted upchirp symbol.

chirp symbols by multiplying them by a downchirp and it then performs an FFT on the de-spread signal. Since the slope of the downchirp is the inverse of the slope of the upchirp, multiplication results in a constant frequency signal, as shown in Fig. 3(a). Thus, taking an FFT on this will lead to a peak in an associated FFT bin. Changing the initial frequency of an upchirp will result in a change in the demodulated signal’s FFT bin peak index which corresponds to the initial change in frequency, as shown in Fig. 3(b). This property is used to convey information. When the sampling rate is equal to chirp bandwidth (BW), frequencies higher than  $\frac{BW}{2}$  will alias down to  $-\frac{BW}{2}$  as shown in Fig. 3(c). This means cyclically shifting in time is equivalent to changing the initial frequency and thus to conserve bandwidth, CSS uses cyclic shifts of the chirp in the time-domain instead of frequency shifts. This means that to modulate the data we just need to cyclically shift the baseline upchirp in time. Note that one can transmit multiple bits within each upchirp symbol. In particular, say the receiver performs an  $N$  point FFT. It can distinguish between  $N$  different cyclic shifts each of which corresponds to a peak in one of the  $N$  FFT bins. Thus, we can transmit  $SF = \log_2 N$  bits within each upchirp symbol, where SF is called the spreading factor.

Based on above explanations, CSS can be characterized by two parameters: chirp bandwidth/sampling rate and spreading factor. Thus, each chirp symbol duration is equal to  $\frac{2^SF}{BW}$  and the symbol rate is  $\frac{BW}{2^SF}$ . Since CSS sends SF bits per symbol, the bitrate is equal to  $\frac{BW}{2^SF} \cdot SF$ . This means increasing SF or decreasing BW decreases the bitrate. Further, the sensitivity of the system depends on the symbol chirp duration and increases with SF and decreases with BW.

## 2.2 Existing Collision Approaches

While existing CSS-based backscatter systems do not support collision decoding, we outline potential approaches to deal with collisions in CSS radio systems, i.e. LoRa, and explore whether they can be adopted for backscatter.

**Using different spreading factors.** One way to enable con-

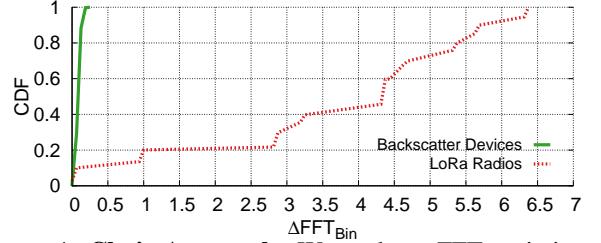


Figure 4: **Choir Approach.** We evaluate FFT variation of chirp symbols when  $BW = 500$  kHz and  $SF = 9$  for both active LoRa radios and backscatter devices.

current transmissions is to assign different spreading factors to each device. There are three problems with using multiple spreading factors in the same network: i) the receiver needs to use multiple FFTs and downchirps with different spreading factors to despread upchirp symbols of different devices, which increases the receiver complexity with the number of concurrent transmissions, ii) in LoRa, different BW and SF can be concurrently decoded without sensitivity degradation, only if the chirp slope is different [24]. Specifically, if two chirp symbols transmitted concurrently with different BW and different SF, which result in the same chirp slope,  $\frac{BW^2}{SF}$  (shown in Fig. 6 as well), the receiver cannot decode their concurrent transmissions. This results in only 19 different BW and SF pairs that could be used concurrently, iii) further, requiring receiver sensitivity better than -123 dBm and bit rates of at least 1 kbps limits these concurrent configurations to only 8, which does not support hundreds of concurrent devices on a 500 kHz band. Note that ignoring the receiver complexity, this approach is orthogonal to our design since we could in principle run multiple concurrent NetScatter networks with the above 8 SF and BW pairs. Evaluating this is not in the scope of this paper.

**Choir [12].** Recent work on decoding concurrent LoRa transmissions leverages the hardware imperfections in radios to disambiguate between multiple transmissions. Specifically, radios have slight variations which result in timing and frequency offsets, which translate to fractional shifts in the FFT indexes. Choir [12] uses these fractional shifts, with a resolution of one-tenth of an FFT bin, to map the bits to each transmitter. However, as demonstrated in [12], in practice this approach does not scale to more than 5 to 10 concurrent devices. To understand this limitation in theory, consider  $N$  concurrent devices. The probability that each of these transmitters has a different FFT peak index fraction, given the resolution of one-tenth of an FFT bin, is equal to  $\frac{10!}{(10-N)!10^N}$ . When  $N$  is 5 this probability is only 30%. Moreover, if any two transmitters use the same cyclic shifted upchirp symbol at the same time, it will result in a collision that cannot be decoded. In the case of LoRa modulation, if there are  $N$  transmitters and assuming each device transmits a random set of bits during each symbol interval, the probability of two transmitters using the same cyclic shift is equal to:

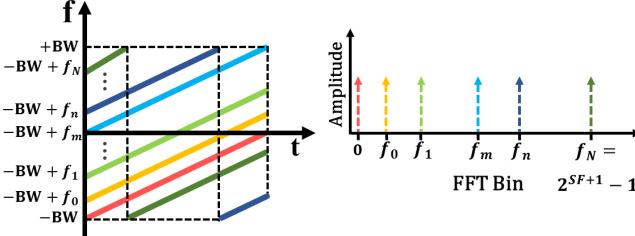


Figure 5: **Bandwidth Aggregation.** Here we use an aggregate bandwidth of  $2BW$  but each device transmits only using  $BW$ . Upchirps with different cyclic shifts shown in different colors. Each upchirp is assigned to a device.

$$1 - \prod_{i=1}^N \left(1 - \frac{i-1}{2^S}\right)$$

which is approximately  $\frac{N(N-1)}{2^{S+1}}$ .

For  $SF = 9$  and  $N = 10$ , this probability is around 9%. This means that there is around 9% probability that within each CSS symbol, two transmitters will use the same upchirp cyclic shift, which the receiver cannot disambiguate. This probability increases to 32% with 20 devices, preventing concurrent decoding of a large number of transmitters.

Moreover, Choir is based on oscillator imperfection causing frequency variation on different devices, and Choir cannot differentiate two concurrent transmissions if both transmissions fall into same FFT bin fraction. Choir uses an active radio system which generates frequencies in 900 MHz band. However, since backscatter systems are designed to consume less power and only generate baseband signals, their output frequency is less than 10 MHz. Now, in the ideal scenario where the same crystal oscillator is used for both radios and backscatter devices, the frequency variation of the backscatter devices is 90 times smaller than radios and can be even less than 1 FFT bin depending on the  $SF$  and  $BW$ . This means a backscatter network cannot use all 10 different FFT bin fractions that Choir have used. Fig. 4 shows CDF of FFT bin variation for our actual backscatter hardware which are recorded over time. This results show that FFT variation is always less than a third of an FFT bin. Thus, Choir cannot enable large concurrent transmissions with backscatter.

In conclusion, the desired solution must satisfy three constraints: 1) ability to differentiate between FFT peaks corresponding to different backscatter devices, 2) ability to associate the FFT peaks to the corresponding devices, and 3) ensure that two devices do not use the same FFT peak at the same time. NetScatter design satisfies all these constraints.

### 3 NetScatter Design

#### 3.1 Distributed CSS Coding

Our approach is to take advantage of low-power and high sensitivity of CSS modulation to design a communication and networking system that enables hundreds of backscatter devices to transmit at the same time.

At a high level, we use a combination of CSS modula-

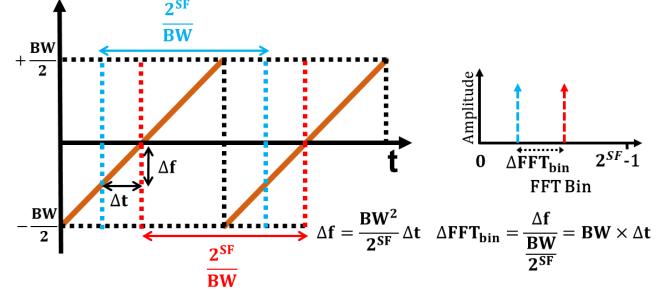


Figure 6: **Timing Mismatch,** in detecting beginning of a chirp symbol and its translation to FFT bin variation.

tion and ON-OFF keying to enable concurrent transmissions. Our intuition is as follows: if we look at the FFT plots of Fig. 3, all the FFT bins except one bin are empty; however these empty bins could be utilized for orthogonal transmissions. While it is difficult to design low-power backscatter devices that can transmit multiple cyclic shifts at the same time, we can leverage all these empty bins by having different devices transmit different shifts and make use of the unused FFT bins. In particular, each device is assigned to a particular cyclic shifted upchirp symbol. It sends data by either sending the upchirp symbol or not sending it, i.e., by using ON-OFF keying of its assigned cyclic shifted chirp. Since, there are  $2^S$  FFT bins, ideally we can support  $2^S$  concurrent transmissions. This modulation will satisfy the above three requirements. The peaks can be differentiated and assigned to their corresponding devices. Moreover, none of them will use the same FFT bin at the same time.

We note the following about our distributed design.

- **Receiver complexity.** The received signal is composed of multiple transmissions. They can be demodulated by despreading with a baseline downchirp multiplication and performing an FFT operation. Then, we can determine the presence and absence of a peak in each FFT bin and find if the corresponding backscatter device is sending ‘0’ or ‘1’. The key point is that the process of despreading and performing FFT, which are the major contributors of the demodulation process and provide a coding gain for each of the backscatter devices enabling them to operate below the noise floor, are being done once and do not depend on the number of concurrent transmissions. This means that the receiver complexity is nearly constant with the number of devices.

- **Throughput gain.** In our approach, ideally there can be as many as  $2^S$  transmissions at each symbol period. Since each backscatter device uses ON-OFF keying over a symbol, their individual data rate is  $\frac{BW}{2^S}$ . Thus, the aggregate network throughput is equal to  $BW$ . In comparison, LoRa have a throughput of  $\frac{BW}{2^S}SF$ . Thus, we can achieve a throughput gain of  $\frac{2^S}{SF}$ , which shows that the gain exponentially increases with the SF value used in the system. This is expected since the number of concurrent devices we can support is an exponential function of SF, i.e.,  $2^S$ .

- **NetScatter and CDMA.** Our distributed CSS coding can be thought of as code-division multiplexing mechanism that is low-power and where each of the  $2^{SF}$  cyclic shifts is in an orthogonal set of codes in a CDMA system. These orthogonal codes are then assigned to  $2^{SF}$  different backscatter devices which enables  $2^{SF}$  concurrent transmissions.
- **Gain in the context of Shannon capacity.** A key gain we are achieving in our design stems from using the power across all the concurrent backscatter devices. Specifically we note that the Shannon capacity of a multi-user network that operates under the noise floor linearly increases with the number of devices. Said differently, the multi-user capacity of an access point network is given as [26],  $C = BW \log_2(1 + \frac{NP_S}{P_N})$ . Here  $BW$  is the channel bandwidth,  $P_N$  and  $P_S$  are the noise and signal power and  $N$  is the number of concurrent devices. At SNRs below the noise floor, the above equation can be approximated as  $\frac{BW}{\ln(2)} \frac{NP_S}{P_N}$ , since  $\ln(1+x) \approx x$  when  $x$  is small. This means that for systems that operate below the noise-floor, the network capacity scales linearly with the number of users. This linear increase stems from the fact that the  $N$  backscatter devices put in  $N$  times more power back to the AP than a single device.
- **Bandwidth Aggregation.** The bitrate achieved by each backscatter device in our distributed design is given by  $\frac{BW}{2^{SF}}$  and the number of concurrent devices is  $2^{SF}$ . Thus, while we can increase the number of devices by increasing  $SF$ , it would decrease the bitrate of each device. Thus, to increase both the bitrate and the number of device we should increase the bandwidth,  $BW$ . Say, we want to support twice the number of devices while maintaining the same bitrate by using twice the bandwidth. This can be achieved in two ways. First, we can use two filters and independently operate two sets of devices across the two bands. This approach requires two different FFTs to be performed independently across the bands. The second approach is to use one aggregate band with twice the bandwidth,  $2BW$ , but use the same  $SF$  and chirp  $BW$  as before and alias down to  $-BW$  whenever the chirp frequency hits the maximum as shown in Fig. 5. To demodulate this signal, we just need to multiply the signal which is composed of the aggregate band by the downchirp and perform  $2 \times 2^{SF}$  FFT operation once. The complexity of this method is lower than the former since there is no need to use filters and separate the bands.

## 3.2 Addressing Practical Issues

### 3.2.1 Timing Mismatch

The above design requires all the backscatter devices to be time synchronized. To understand why, consider two consecutive upchirps being sent by a device, as shown in Fig. 6. Now say that we demodulate the signal in these two timing durations, shown in blue and red, we will get different FFT peak locations. Specifically, with a  $\Delta t$  time difference be-

tween these durations, the corresponding FFT bin peak location would change by,  $\Delta FFT_{bin} = \Delta t BW$ . When this change is greater than a single FFT bin, backscatter devices that are assigned to consecutive cyclic shifts interfere with each other and hence cannot be decoded. Thus, all the devices should be time synchronized. In our design the access point sends a query message telling devices to transmit concurrently. The devices use this query to synchronize and respond concurrently. First, we explain the sources of time delay in our system and then we explain our solution. There are multiple factors that can contribute to time delays introduced in practice and can be different for different backscatter devices.

- **Hardware delay.** Unlike Wi-Fi devices which use much higher clock frequencies for processors, backscatter devices use low-power microcontrollers (MCU) that can introduce a variable delay into the system. For backscatter devices, the source of these hardware delay variations come from the time the envelope detector receives the query message from the access point, communicates it to the MCU and then the device backscatters the chirp. As we show in §4.2, this hardware delay variations can be as high as  $3.5 \mu s$ , which can translate to more than one FFT bin at 500 kHz bandwidth.

- **Propagation delay and multipath.** Since backscatter devices can be at different distances to the access point, their time of flight (TOF) can be different. However, since our target application is for whole-home or whole-office sensing, the propagation distance is less than 100 m which translates to a  $ToF < 666ns = \frac{2 \times 100}{3 \times 10^8}$  and corresponds to only a 0.33 FFT bin change, assuming a bandwidth of 500 kHz. The multipath delay spread for indoor environments is between 50 to 300 ns [23, 11]. For 500 kHz, this delay spread translates to less than 0.15 FFT bin change, which is negligible.

**Our solution: Bandwidth-based cyclic-shift assignment.** Hardware delay variations over time are hard to correct for. As described above, by nature of operating on MCUs and other low-power computational platforms, these devices have a hardware delay variation over time that changes between packets. Our solution to this problem is to put a few empty FFT bins adjacent to each FFT bin assigned to a device. That is, if FFT bin  $i$  is assigned to a device, the adjacent  $SKIP - 1$  FFT bins are empty and not assigned to any device. This can be done by using only every  $SKIP^{th}$  cyclic shift of the chirp. This ensures that the hardware delay does not result in adjacent devices interfering with each other.

Achieving such an assignment requires us to answer the following key question: how do we pick the value  $SKIP$ ? As described earlier, given the hardware delay variation  $\Delta t$ , the shift in the number of FFT bins is  $\Delta t BW$ . This means that there is a trade-off in our system regarding the total network throughput, bitrate for each device and sensitivity. In particular, increasing  $BW$  increases the number of FFT bins that have to be left empty and decreases the total network throughput. On the other hand, decreasing  $BW$  reduces the

Table 1: NetScatter Different Modulation Configurations, with maximum time/freq. mismatch that can be tolerated.

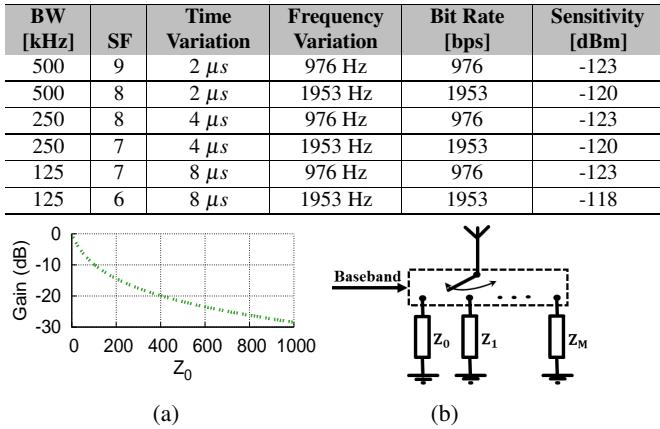


Figure 7: Power Adjustment for Backscatter. (a) gain normalized to maximum power as a function of  $Z_0$  impedance and (b) switch network to support multiple power levels.

number of FFT bins but decreases the bitrate per device with the same  $SF$ . To compensate for the decreased device’s bitrate, we can decrease the  $SF$ . Note that, we can choose total bandwidth, chirp  $BW$  and  $SF$  of the system by considering the hardware delay variations, required bitrate per device, sensitivity for each device and total number of devices. For our implementation, we pick the same total bandwidth and chirp  $BW$  of 500 kHz and  $SF = 9$  which supports around 1 kbps (976 bps) bitrate at the devices while ensuring that the number of empty bins between devices,  $SKIP$ , is two.

### 3.2.2 Frequency Mismatch

The devices experience frequency offsets because of hardware variations in the crystals used in their oscillators. As explained in §2.1, change in frequency translates to FFT bin change of the demodulated device packet. This again, causes one device to be misinterpreted as other device. Considering a bandwidth of  $BW$  and spreading factor of  $SF$ , the frequency difference between FFT bins is equal to  $\frac{BW}{2SF}$ . This means that a  $\Delta f$  frequency offset results in a change in the FFT bin of  $\Delta FFT_{bin} = \frac{2^{SF}\Delta f}{BW}$ . Therefore either increasing the spreading factor  $SF$  or decreasing the  $BW$  can increase the shift in the FFT bin. Crystals’ frequency tolerance can be as high as 100 ppm [2]. Since backscatter devices run at a few MHz frequencies, this frequency variation translates to less than one FFT bin for the bandwidths and spreading factors in this paper which makes it negligible for our backscatter network.

Table 1 shows the timing and frequency mismatch that can be tolerated for different modulation configurations. As can be seen, there are multiple options for achieving the same bitrate and sensitivity. These options will result in different tolerable timing and frequency mismatch, requiring a different  $SKIP$  value; this is validated using experiments in §4.2.

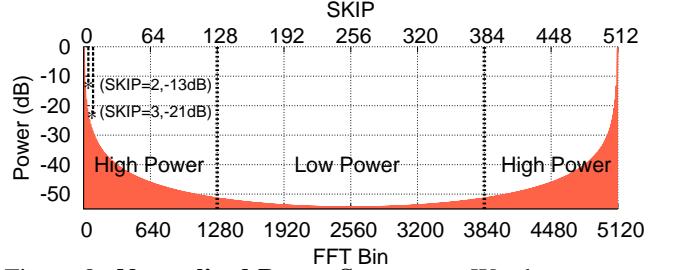


Figure 8: Normalized Power Spectrum. We show power spectrum of an upchirp multiplied by a baseline downchirp in FFT domain. This plot shows the main lobe and side lobes of a single chirp transmission. We assign devices to high and low power regions based on their power level.

### 3.2.3 Near-Far Problem

Since our networks are designed to work in below-noise conditions, we need to address the near-far problem in our decoding process at the receiver. Specifically, to account for the residual timing and frequency offsets, a CSS receiver has to achieve a sub-FFT bin resolution. To do so without increasing the sampling rate, the receiver uses zero-padding which adds zeros at the end of the time domain samples of the single chirp [12]. Zero-padding operation in the time domain is effectively a multiplication operation with a pulse which translates to convolution with a sinc function in the FFT domain. This makes it easier to locate the FFT peak location. However, convolving with a sinc function introduces side lobes as shown in Fig. 8. Assume that there are two devices with cyclic shifts  $C_1 = 0$  and  $C_2$ . If the power of  $C_2$  is lower than power of  $C_1$ ’s side lobes, it cannot be decoded.

**Our solution.** To address this issue, we propose two techniques that work together to increase our dynamic range.

*Coarse-grained power-aware cyclic shift assignment.* Our intuition here is as follows: Fig. 8 suggests that we should assign adjacent FFT bins to devices that have a small SNR difference. In particular, when  $SKIP$  is 2, the neighboring backscatter device will be drowned by the power of the higher SNR device if its power is lower than 13.5 dB of that of the high SNR device. Further, it shows that the side-lobe power of a high SNR device decreases as we go to farther FFT bins. Thus, we need to ensure that a lower SNR device has to correspond to FFT bins that are farther from the FFT bins corresponding to higher SNR devices. This ensures that the side-lobes of the high-SNR device do not affect the decoding of the low-SNR devices. Specifically, we assign different cyclic shifts to different devices at association phase to ensure that the FFT bins corresponding to the lower-SNR devices are close to each other and are far from higher-SNR devices. To do this, the AP computes the signal strength of the incoming device in the association phase (see §3.3.2) and assigns its cyclic shift based on its signal strength and also the strengths of the devices already in the network.

We run simulations to understand the benefits of this al-

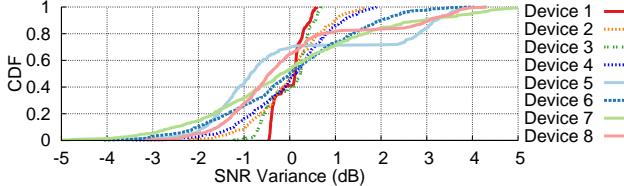


Figure 9: **Backscatter Devices SNR Variance.** CDF of SNR variance of backscatter devices in an office environment, when people were walking around, over 30 mins.

location. Specifically, we assign two devices to FFT bins 2 and 258, with  $SF = 9$  and  $BW = 500 \text{ kHz}$ . To be realistic, we added Gaussian frequency mismatch with variance of 300 Hz to each device to account for timing and frequency mismatches between them. We change the power of the second device and measure the bit error rate (BER) for the first device. Fig. 12 shows the BER over  $10^4$  symbols, for different power differences between the two devices. As can be seen, the BER remains unaffected even when the second device is around 40 dB stronger than the first device. This shows that our power-aware allocation can in theory tolerate power difference of 40 dB between devices. In practice however this is a little lower at 35 dB (see §4.3).

*Fine-grained self-aware power-adjustment.* While the above assignment is determined at association, mobility in the environment and fading will change the SNR of each of the devices over time (see Fig. 9). To address this, each device adjusts its power over time using the signal strength of the query message from the AP, using three different levels. We define the maximum power of the device as 0 dB power gain. First, during association, we consider two cases for the associating device. If it sees a low received signal strength for the AP’s query packet, it sets its power gain to the maximum. Otherwise, it sets its gain to the middle level. This gives the higher signal strength backscatter devices leeway to both increase and decrease their power, after association. The AP uses the resulting backscatter signal strengths during association to assign a corresponding cyclic shift. The backscatter devices use the signal strength at association as a baseline and either increase or decrease their power gains for the rest of the concurrent transmissions, i.e., if the signal strength for the AP’s query message increases (decreases), the backscatter devices decrease (increase) their power gain. If the device cannot meet its expected SNR requirements given its limited power levels and assigned cyclic shift, it does not join the concurrent transmissions. If this happens more than twice, the backscatter device re-initiates association after which the AP reassigns the cyclic shifts to account for the new significantly different power value (see §3.3.2).

The key question however is: how can a low-power backscatter device change its transmission power gain? This is interesting since power adaptation has not been used before in the network of backscatter devices. In backscatter, the transmit power gain,  $Gain_{power}$ , is equal to  $\frac{|\Gamma_0 - \Gamma_1|^2}{4}$ . Here

$\Gamma_0$  and  $\Gamma_1$  are reflection coefficients for switching between two impedance value,  $Z_0$  and  $Z_1$ . Backscatter hardware is designed to maximize the difference between reflection coefficients to maximize their transmission power. This corresponds to  $Gain_{power} = 0 \text{ dB}$ . One way to achieve this is to switch between extreme impedance values,  $Z_0 = 0\Omega$  and  $Z_1 = \infty\Omega$ . To achieve power adaptation, in contrast, we pick impedance values that correspond to multiple power settings. In particular, as shown in Fig. 7a, instead of switching from  $Z_0 = 0\Omega$ , we switch from intermediary impedances and hence achieve lower power gains. Our hardware implementation achieves three power gains of 0 dB, -4 dB and -10 dB to achieve power adaptation. Note that [25] uses a similar circuit structure as Fig. 7b to cancel higher order harmonics. We instead design this circuit structure to control the power.

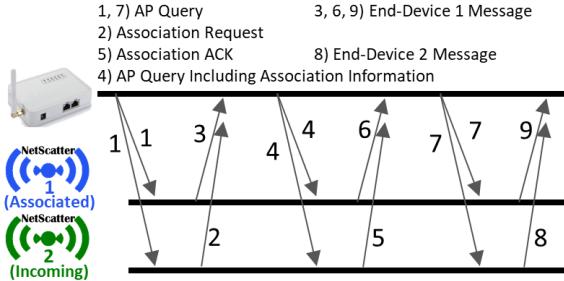
*Design tradeoff.* Readers might wonder if reducing the power of high SNR devices would decrease the network throughput, since high SNR devices in traditional LoRa backscatter designs can achieve a higher bitrate. In contrast, by reducing their power we are enabling a large number of concurrent transmissions with a fixed bitrate. Thus, we are encouraging concurrency by reducing the bitrate of high SNR devices. §4.4 compares the results for NetScatter with one where each backscatter device uses rate adaptation to pick its ideal bitrate, while transmitting alone using LoRa backscatter [25]. The results show that the network throughput and latency gains due to large scale concurrency outweigh the reduction in the power for high SNR devices.

### 3.3 NetScatter Protocol & Receiver Details

Putting it together, the AP transmits an ASK modulated query message which is used to synchronize all the participating concurrent devices. This message conveys information about cyclic shift assignment which are based on the devices’ signal strength at the AP. The devices measure the query message’s signal strength using the envelope detector and use it to fine-tune their transmit power gain. In the rest of this section, we describe various protocol details required to make our design work in practice. Note that our focus in the protocol design is about scheduling a set of concurrent transmissions. Typically networks could have more devices than concurrent transmitters supported by our design. Since the AP knows the duty-cycle of each device from the association phase (see §3.3.2), it can i) assign the cyclic shifts and ii) schedule the devices involved in concurrent transmissions.

#### 3.3.1 Link-layer Backscatter Packet Structure

Similar to LoRa, the device packet starts with upchirp and downchirp preambles. They are designed to serve two purposes: i) finding the start of the packet and ii) detecting the transmissions. We emphasize here that the device transmits the same assigned cyclic shift for both up-



**Figure 10: NetScatter Network Association Process.** We show the association process of an incoming NetScatter device (#2) to the network, while there are existing devices associated with the network (i.e., device #1).

chirps and downchirps in the preamble as well as the payload. The preamble consists of six upchirps followed by two downchirps. This is then followed by the payload and the checksum. We note that in our design, all the devices send their preambles concurrently. This reduces the overhead of transmitting preambles for each device, which in turn increase the end-to-end throughput gain achieved by NetScatter. The AP uses the above structure to achieve two goals.

i) *Finding the exact packet start.* We use the downchirp in the preamble to find the start of the packet transmission. Specifically, we use the middle point between an upchirp and downchirp and switch by six upchirp symbols to the left to find the packet beginning. We suspect that the LoRa preamble has a downchirp for this exact purpose. We note that in our case, since the upchirp and downchirp in the preamble from each of the devices uses the same cyclic shifts, they are symmetric around the middle point and hence the same algorithm for estimating the packet beginning is applied.

ii) *Detecting and decoding each concurrent transmitter.* Now that we found the packet start, we need to find out which transmitters are in the network. To do so, for each preamble symbol, we demodulate it and look at the peaks in FFT domain. If there is an FFT peak in the demodulator output which repeats in all the preamble symbols, we conclude that the device corresponding to that cyclic shift is sending data. After finding current devices in the network, we compute the average power over the six preamble symbols for each device. This average power is used as a threshold to demodulate the payload of each device. In particular, if the power of the device's FFT peak for each payload symbol is more than half this average, we interpret that as 1 and 0 otherwise.

### 3.3.2 Network Association

Say the network already has  $N$  devices associated to the AP and the  $N + 1^{\text{th}}$  device wants to join the network. A naïve approach is to periodically dedicate time periods for association. This however can lead to high association delays depending on the frequency of the association periods. Our approach instead is to reserve  $N_{\text{assoc}}$  cyclic shifts and the cor-

Preamble [16]	Packet Length [8]	Group ID [4]	Device ID [8] (Optional)	Cyclic Shift [8] (Optional)	CRC
---------------	-------------------	--------------	--------------------------	-----------------------------	-----

**Figure 11: Structure of AP’s Query Message.**

responding FFT bins for association and use the rest for communication. In other words, all the devices transmit at the same time but the ones who want to enter the network transmit with the  $N_{\text{assoc}}$  association cyclic shifts.

To address the near-far problem, we reserve two cyclic shifts, one in high-SNR and the other one in the low-SNR cyclic shift regions. The incoming device would choose which association region to transmit based on the signal strength of the AP’s query message, calculated using the envelope detector. However to account for the hardware delay variations, as before, we skip two cyclic shifts to ensure that the association packets from the devices can be decoded and won’t interfere with communication cyclic shifts. Finally, to support scenarios where more than one device want to associate at the same time, one can use Aloha protocol with binary exponential back-off in the association process. Our deployment does not implement this option and turns ON the backscatter devices one at a time and runs the network only after all the devices are associated.

After the incoming device sends its packet to the AP in association process using the association cyclic shifts, the AP computes its signal strength and decides which cyclic shift and timing schedule it should be assigned to. The AP piggybacks these assignments in its query messages.

### 3.3.3 AP Query Message

Fig. 11 shows the ASK-modulated query message that the AP sends. The message has a group ID which identifies the set of 256 devices that should concurrently transmit. In our implementation, since there are only 256 devices, we set this group ID to 0. In a larger network, the AP can assign different sets of devices to different groups depending on their signal strengths, i.e., devices that have a similar signal strength are grouped into the same group to enable concurrent transmissions while further minimizing the near-far problem.

This is then followed by an optional association response payload that assigns an 8-bit network ID and a 8-bit cyclic shift. Note that prior LoRa backscatter designs are request-response systems that query each backscatter device sequentially and need most of the fields in Fig. 11 other than the group ID and cyclic shift assignment. Since these additional 12 bits is transmitted using 160 kbps ASK downlink, the overhead is negligible compared to the 1 kbps backscatter uplink. Finally, we note that if the AP is unable to assign a new device given the existing assignments, the AP updates the cyclic shift assignments for all the devices in the network. It does so by transmitting the identifier for one of the the 256! orderings, which requires  $\log_2(256!)$  ( $\leq 1700$ ) bits. This occupies less than 11 ms using our 160 kbps downlink.

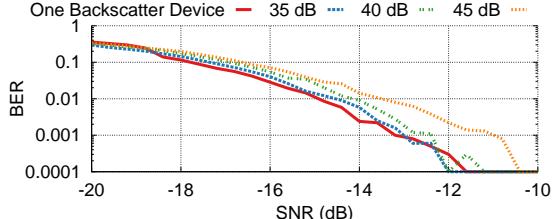


Figure 12: **Near-Far BER Results.** We show the effect of the second device’s power on the first device’s BER vs. SNR for different ratios of the second device’s to first device’s power with power aware cyclic shift assignments.

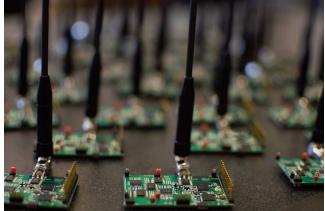


Figure 13: **Our Backscatter Devices.** They are arranged closely for this picture. They are spread out across more than ten rooms in our deployment.

### 3.3.4 Network protocol

Fig. 10 summarizes our network protocol. First the AP broadcasts its query. Device 1, which is already associated to the network receives the query and sends its data using its assigned cyclic shift after performing any necessary power control. Concurrently, device 2 sends a *Association Request* using one of the  $N_{\text{assoc}}$  cyclic shifts. The AP receives these two messages and broadcast another query which includes association information for device 2. Upon receiving this query, Device 1 continues to send its data, however, device 2 extract cyclic shift assignment from the query and then transmits *Association ACK* to the AP in the assigned cyclic shift. If AP receives *Association ACK*, it adds device 2 to associated devices. Otherwise, it will repeat the association information in the following queries. After association, each device uses its assigned cyclic shift for sending data.

## 4 Evaluation

### 4.1 Hardware implementation

**COTS Implementation.** Our COTS hardware shown in Fig. 13 consists of RF section and baseband section, both implemented on a four layers FR4 PCB. On RF receive side, we implemented envelope detector similar to [18] but at 900 MHz and it has a sensitivity of -49 dBm to receive downlink query messages from AP.<sup>1</sup> RF transmit side consists of five ADG904 [3] switches cascaded in three levels to build an

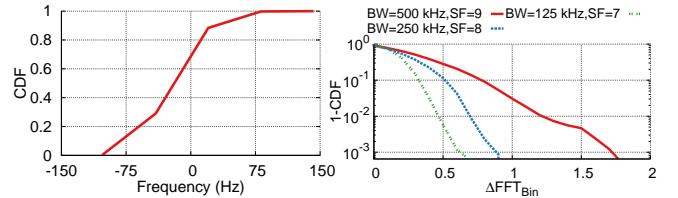


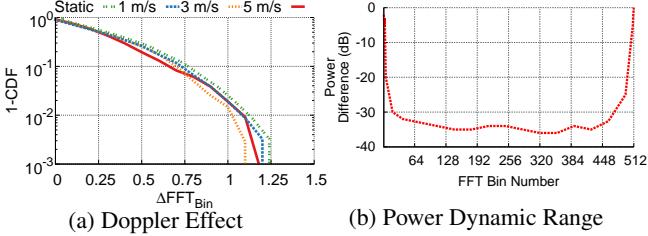
Figure 14: **Frequency Offset FFT Bin Variation.** (a) frequency offset of backscatter devices, and (b) effect of residual time and frequency offset for different configurations.

impedance switch network for backscatter, power gain control and also switching between transmit and receive modes. Our backscatter device uses a 2 dBi whip antenna to transmit packets and receive query messages in the 900 MHz ISM band. The baseband side is implemented using an IGLOO nano AGLN250 FPGA [1] and an MSP430FR5969 [5]. We generate CSS packets on the FPGA and output real and imaginary components of the square wave signal to the backscatter switch network. The envelope detector is controlled by the MCU. Downlink receiver algorithm is implemented on MCU. To be resilient to self-interference caused by the AP’s single-tone, the baseband at the backscatter device shifts the AP’s signal by 3 MHz. Note that the COTS implementation is for prototyping and proof-of-concept; an ASIC is typically required to achieve the orders of magnitude power benefits of backscatter communication.

**IC Simulation.** We designed and simulated an IC for our backscatter device using TSMC 65nm LP process. It consists of four blocks with total power consumption of 45.2  $\mu W$ : i) An envelope detector that demodulates the APs ASK query messages and consumes less than 1  $\mu W$ . ii) Baseband processor for processing and extracting AP data from envelope detector, interfacing with sensors and sending the chirp specifications and sequence of data to chirp generator is handled by this block consuming 5.7  $\mu W$  of power. iii) A chirp generator that takes SF, BW, cyclic shift assignment and data sequence from the baseband processor to generate the sequence of ON-OFF keying chirps. We used Verilog code to describe the baseband signal’s phase behavior and generate assigned cyclic shift with required frequency offset. We used Synthesis, Auto-Place and Route (SAPR) to simulate Verilog code on chip. The power consumption of this block is 36  $\mu W$ . iv) A Switch network which is composed of three resistors that are connected to NMOS switches to generate backscatter signal with three power gain levels. Note that since these resistors and NMOS switches consume minimal area, more power gain levels can be added at almost no cost. The power consumption of the switch network is 2.5  $\mu W$  with 3 MHz frequency offset.

**Reader Implementation.** We implement the reader on the X-300 USRP software-defined radio platform by Ettus Research [8]. We use a mono-static radar configuration with

<sup>1</sup>Note that since ASK-modulated AP query received by backscatter device experiences one-way path loss, its required sensitivity is only -44 dBm in contrast to the -120 dBm sensitivity for the backscatter signals.



**Figure 15: Doppler Effect and Power Dynamic Range Evaluation.** We evaluate (a) Doppler effect, and (b) we show power difference between two concurrent transmissions at different locations of FFT domain. One transmission is fixed and the other is sweeping across different chirp symbols.

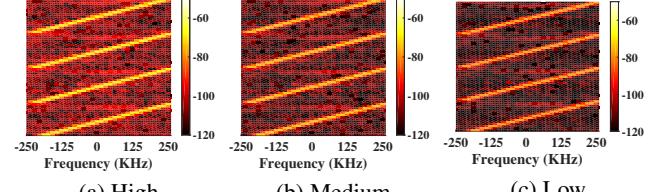
two co-located antennas separated by 3 feet. The transmit antenna is connected to a UBX-40 daughterboard, which transmits the query message and the single-tone signal. The USRP output power is set at 0 dBm and we use an RF5110 RF power amplifier [6] to amplify the transmit signal to 30 dBm. The receiver antenna is connected to another UBX-40 daughterboard, which down-converts the NetScatter packets to baseband signal and samples it at 4 Msps.

## 4.2 Frequency and Timing Mismatch

*Measurements 1: Hardware frequency variations.* We measure the frequency offsets of our hardware by recording thousand packets for each device. Using the method described in §3.3.3, we compute the frequency offset for the 256 backscatter devices in our network deployment which we show in Fig. 14a. The variations of backscatter devices are less than 150 Hz which is nearly 0.15th of one FFT bin when  $BW = 500\text{kHz}$  and  $SF = 9$ . Therefore, our system is not affected by frequency variation of different devices.

*Measurements 2: Timing offsets.* Next, we characterize how the timing offsets affects  $\Delta FFT_{bin}$ . This helps us understand how many empty cyclic shifts,  $SKIP - 1$ , we need to put for each occupied cyclic shift. To do this, we setup a wireless experiment sending query messages from the AP and receiving transmissions from the backscatter devices deployed in our system. By decoding these transmissions and comparing the received cyclic shifts with what we have programmed the devices to send, we can find the  $\Delta FFT_{bin}$  for each device; this measurement is a combination of both timing and the small frequency variations on the hardware.

Fig. 14b shows residual  $\Delta FFT_{bin}$  for backscatter devices. The plots show that the  $\Delta FFT_{bin}$  is considerable. This is because in backscatter devices, the energy detector receives the amplitude modulated query message and sends interrupt to initiate backscatter transmission. Both these steps add to the timing variations. Specifically, the hardware delay variation comes from variation in receiving query message and initiating the transmission on FPGA which can vary from packet to packet. In our deployment in §4.4 with backscatter devices, we use  $BW=500\text{ kHz}$ ,  $SF=9$  and leave one FFT bin



**Figure 16: Spectrogram of Backscattered Signal at the Different Power Levels.**

between occupied cyclic shifts ( $SKIP = 2$ ). This translates to supporting 256 devices with an aggregate throughput of around 250 kbps and bitrate per tag of around 1 kbps.

*Measurements 3: Doppler effects.* Other than hardware frequency offsets, Doppler effect can cause changes in frequency as well. However, the effect of it will be much less than 1 FFT bin,  $\frac{BW}{2SF}$ , for most cases. As an example, assume a backscatter device is moving with a speed of 10 m/s. Considering the carrier frequency is 900 MHz, the doppler effect induced frequency change would be 30 Hz which is much less than 1 kHz, the FFT bin frequency, assuming  $BW=500\text{ kHz}$  and  $SF=9$ . To confirm this, we run various mobility experiments where a subject holds a backscatter device and moves with different average speeds which we measure using an accelerometer. We receive transmissions from the device and compute the  $\Delta FFT_{bin}$  for different motion scenarios. Fig. 15a shows  $\Delta FFT_{bin}$  for various speeds, which confirms that these speeds do not have an effect on  $\Delta FFT_{bin}$ .

## 4.3 Near-Far Problem

*Measurements 1: Power-aware cyclic shift assignment.* As mentioned in §3.2.3, we assign cyclic shifts to devices depending on their signal strength values. To evaluate the effectiveness of this technique, we run experiments with two devices where one of them transmits at a high power (equivalent to being near the AP) with a cyclic shift corresponding to the beginning of the FFT spectrum. Then, we sweep the cyclic shift of the second device from small FFT bin difference cyclic shifts to high FFT bin difference ones. At each of the cyclic shifts, we decrease the power of the second device using an attenuator up to when it has packet error rates less than one percent. Fig. 15b shows the maximum power difference that can be tolerated between these two devices versus the assigned FFT bin difference. As can be seen, as we go further in FFT bin difference, we can tolerate more power difference between the two devices. Note that, because of aliasing Fig. 15b is symmetric around the center. The maximum happens in middle and is equal to 35 dB. This is the dynamic range that our system can support in practice. We also note that when the second device is assigned to an FFT bin 2 cyclic shifts away from the first device, it can be up to 5 dB below the latter and still correctly decoded. This means there is an in-built 5 dB dynamic range resilience to channel

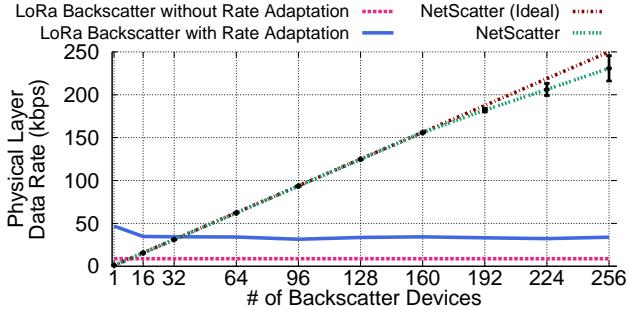


Figure 17: **Network Physical Rate.** We evaluate NetScatter network physical rate and compare it with other schemes.

variations between devices that have close cyclic shifts.

*Measurements 2: Self-aware power-adjustment.* The second method to address the near-far problem and also increase the dynamic-range is power adjustments at the devices using the signal strength of the AP’s query message. To evaluate this, we first measure how well we can adjust power on the devices. We evaluate its efficacy in practical deployments. **We use three different backscatter impedance values** to be able to transmit packets in three different power gains. Fig. 16 shows the spectrum of backscattered signal at different power levels. These plots show that the hardware creates spectrum that is clean and does not introduce noticeable nonlinearities into the backscattered signal. Furthermore, we can achieve three different power levels: 0, -4, and -10 dB.

#### 4.4 Network Deployment

We evaluate three key network parameters:

- **Network PHY bit-rate.** This is the bitrate achieved across all the devices during the payload part of the packet.
- **Link-layer data rate.** This is the data rate achieved in the network which is defined as the data rate for sending useful payload bits, after considering overheads including the AP’s query message and the preamble of the packet transmission.
- **Network latency.** This is the latency to get the payload bits from all the backscatter devices in the network.

We compare three schemes: i) LoRa backscatter [25] where all devices use a fixed bitrate of 8.7 kbps, ii) LoRa backscatter with rate adaptation where each device uses the best bitrate given its channel conditions and iii) NetScatter. Note that the authors of [25] did not publicly release the code and so, we replicate the implementation adding the missing details and using  $BW = 500 \text{ kHz}$  and  $SF = 9$ . We also note that [25] is not designed to work with more than one to two users. Here, we use query-response design with scheduling when there are more users where the AP queries each device. While LoRa backscatter does not support rate adaptation, we wanted to compare with an ideal approach that maximizes the bitrate of each device by picking the optimal  $SF$  and  $BW$ .

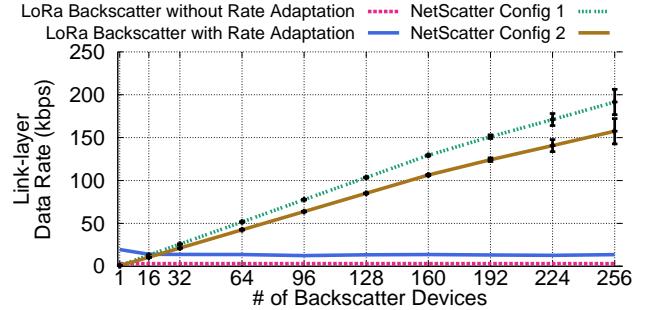


Figure 18: **Link-layer Data Rate.** We evaluate link-layer data rate for NetScatter and compare it with other schemes.

To do so, we measure the signal strength from each of the backscatter devices and compute the bitrate using the SNR table in [4]; this is the ideal performance a single-user LoRa backscatter design achieves with rate adaptation.

**Network PHY bit-rate.** We set each device bitrate to 976 bps,  $BW_{agg} = 500 \text{ kHz}$ ,  $SF = 9$  and a payload size of five bytes. We deploy 256 backscatter devices across the floor of an office building with more than ten rooms. Fig. 1 shows our deployment in an office. Fig. 17 shows the results of network physical rate for our backscatter network deployment. The plot highlights the following key observations.

- The network data rate scales with the number of concurrent backscatter devices. When the number of concurrent devices is less than 128, the variance in the throughput is small. This is because in these scenarios effectively the backscatter devices are separated from each other by more than 2 cyclic shifts ( $\text{SKIP} \geq 3$ ). As a result, the devices do not interfere with each other and hence can concurrently operate. As we increase the concurrent devices to 256, we are pushing the system to its theoretical limit (with  $\text{SKIP} = 2$ ) and thus, we see larger variances in the network data rate.
- With 256 backscatter devices, NetScatter increases the PHY bit-rate by 6.8x and 26.2x over LoRa backscatter with and without rate adaptation. The gains are lower with the ideal rate adaptation since with rate adaptation high-SNR devices could pick the maximum LoRa bitrate of 32 Kbps.

**Link-layer data rate.** While the above plots measure the data rate improvements for the message payload, it does not account for the end-to-end overheads including preambles and the AP’s query message to coordinate the concurrent transmissions. To see the effect of the AP query packet overhead for NetScatter, we consider two configurations.

- *NetScatter Config 1.* In this scenario the cyclic shifts are all assigned during the association phase and the AP query packet coordinating the concurrent transmissions is 32 bits long without the optional fields in Fig. 11.
- *NetScatter Config 2.* In this scenario, the AP query packet contain cyclic shift assignments for all the devices in the network and has a length of 1760 bits.

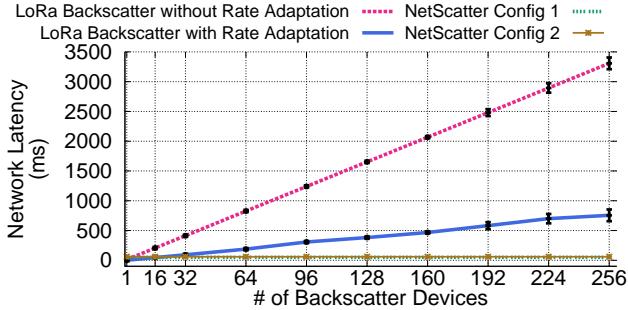


Figure 19: **Network Latency.** We evaluate the latency of NetScatter and compare it with other schemes.

The above two configurations represent the two extremes of our deployment. We set the backscatter payload and CRC to 40 bits and use the total 8 upchirps and downchirps for preamble. For LoRa backscatter which queries each individual device sequentially, the AP query is 28 bits long.

Fig. 18 shows that the gains at the link-layer are higher for NetScatter over LoRa backscatter without and with rate adaptation by 61.9x (50.9x) and 14.1x (11.6x) respectively for config#1 (#2). This is because, in NetScatter, the added overhead of devices’ preambles happen once and at the same time for all devices. But the other schemes need to do TDMA which means that sending preamble will not happen concurrently for all devices and these have to be sent individually for each backscatter device since in traditional designs the AP querying each of them sequentially. Further, in LoRa backscatter which queries sequentially, the AP query message is transmitted once for each device in the network versus being transmitted once for all the devices in our design. Finally, since the downlink uses ASK at 160 kbps, the overhead of transmitting 1760 bits in config#2, while reducing the link-layer data rate over config#1, is still low because the backscatter links can only achieve a much lower bitrate.

**Network latency.** Finally, Fig. 19 shows that NetScatter has a latency reduction of 67.0x (55.1x) and 15.3x (12.6x) over prior LoRa backscatter without and with rate adaptation respectively in network config#1 (#2). This is the key advantage of using concurrent transmissions in low-power backscatter networks. It is noteworthy that since the downlink AP query bit-rate is 160 kbps, AP query duration is negligible compared to duration of backscatter devices’ preamble for prior backscatter methods and also for config#1. For config#2, the AP query duration is significantly higher than the config#1. However, the total duration is still dominated by the backscatter payload + CRC and preamble. As a result, AP query is not the dominant factor in link-layer latency.

## 5 Related Work

Recent systems use backscatter with Wi-Fi signals [10, 29, 18], have a receiver sensitivity of only -90 dBm and hence have a limited range and cannot work across rooms unless

the RF source is placed close to the backscatter tag [19, 18]. LoRa backscatter [25] can achieve long ranges by generating LoRa-compliant packets at the backscatter device. pLoRa [22] backscatters ambient LoRa signals in the environment in contrast to the single tone used as the RF source in NetScatter as well as [25]. We note that all SemTech LoRa chipsets have the capability in software to transmit single tone signals. All these prior long range systems are evaluated in a network of only 1–2 devices and propose to use time-division to support multiple backscatter devices. In contrast, our design enables large-scale concurrent transmissions and can achieve much higher link-layer data rates as well as lower latencies. We also note that these long range backscatter systems [25, 22] claim a kilometer range in outdoor scenarios such as open fields. This however requires placing the RF source close to the backscatter devices. In indoor environments where the signal propagates through walls and the RF source is not placed close to the backscatter devices, our network operational range across ten different rooms is consistent with these prior work. Finally, we note that while prior work [25, 22] decodes the backscatter signal on Semtech LoRa chipsets, our distributed CSS protocol is decoded on a software radio. We however note that SemTech LoRa SX1257 [7] chipsets provide I-Q samples and hence our approach could also be implemented on these off-the-shelf chipsets together with a low power FPGA for baseband processing; this however is not in the scope of this paper.

Finally, recent work on decoding concurrent transmissions from RFID tags, does not achieve the long range operations and below-noise operations of CSS based systems. Buzz [28], LF-Backscatter [13], and others [14, 21, 16] leverage the differences in the time domain signal transitions and changes in the constellation diagram to decode multiple RFIDs. However, the number of concurrent transmissions in the above designs is limited — the latest in this line of work, Fliptracer [16], can reliably decode up to five concurrent RFID tags. Further, these systems were tested with ranges of 0.5 to 6 feet [28, 13, 16] and in the same room. Finally, receiver sensitivity of even battery-powered backscatter tags for RFID EPC-GEN2 readers is around -85 dBm. So it cannot support the long ranges and whole-home deployments that CSS modulation based backscatter achieves.

## 6 Conclusion

We present a new wireless protocol for backscatter networks that scales to hundreds of concurrent transmissions. To this end, we introduce, distributed chirp spread spectrum coding, which uses a combination of chirp spread spectrum (CSS) modulation and ON-OFF keying. Further, we address practical issues including near-far problem and timing and frequency synchronization. Finally, we deploy our system in an indoor environment with 256 concurrent devices to demonstrate its throughput and latency performance.

## References

- [1] Igloo nano fpga datasheet by, 2015. [https://www.microsemi.com/document-portal/doc\\_download/130695-igloo-nano\\_low-power-flash-fpgas-datasheet](https://www.microsemi.com/document-portal/doc_download/130695-igloo-nano_low-power-flash-fpgas-datasheet).
- [2] Xrcha-f-a series, 2015. <https://www.murata.com/~media/webrenewal/applications/wireless/transceiver/f1rj078-dashx>.
- [3] Adg904 datasheet by analog devices, 2016. <http://www.analog.com/media/en/technical-documentation/datasheets/ADG904.pdf>.
- [4] Sx1276 datasheet by semtech, 2016. <https://www.semtech.com/uploads/documents/sx1276.pdf>.
- [5] Msp430fr5969 datasheet by ti, 2017. <http://www.ti.com/lit/ds/symlink/msp430fr5969.pdf>.
- [6] Rf5110 rf power amplifier, 2018. <http://www.rfmd.com/store/downloads/dl/file/id/30508>.
- [7] Sx1257 datasheet by semtech, 2018. <https://www.semtech.com/uploads/documents/DS-SX1257-V1.2.pdf>.
- [8] Usrp x-300, 2018. <https://www.ettus.com/product/details/X300-KIT>.
- [9] A. Berni and W. Gregg. On the utility of chirp modulation for digital signaling. *IEEE Transactions on Communications*, 21(6):748–751, Jun 1973.
- [10] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti. Backfi: High throughput wifi backscatter. SIGCOMM ’15.
- [11] D. M. Devasivatham. Time delay spread measurements of wideband radio signals within a building. *Electronics Letters*, 20(23):950–951, 1984.
- [12] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan. Empowering low-power wide area networks in urban settings. SIGCOMM ’17.
- [13] P. Hu, P. Zhang, and D. Ganesan. Laissez-faire: Fully asymmetric backscatter communication. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM ’15.
- [14] P. Hu, P. Zhang, and D. Ganesan. Leveraging interleaved signal edges for concurrent backscatter. In *Proceedings of the 1st ACM workshop on Hot topics in wireless*, pages 13–18. ACM, 2014.
- [15] 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*.
- [16] M. Jin, Y. He, X. Meng, Y. Zheng, D. Fang, and X. Chen. Flip-tracer: Practical parallel decoding for backscatter communication. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom ’17.
- [17] 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*.
- [18] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith. Passive wi-fi brings low power to wifi transmissions. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI’16)*. <https://nsdi16.usenix.org/works/adG904.pdf>.
- [19] M. Kotaru, P. Zhang, and S. Katti. Localizing low-power backscatter tags using commodity wifi. In *CoNext’17*.
- [20] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient backscatter: Wireless communication out of thin air. SIGCOMM ’13.
- [21] J. Qiu, M. Li, and Y. Zheng. Come and be served: Parallel decoding for cots rfid tags. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 500–511. ACM, 2015.
- [22] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson. PLoRa: Passive long-range data networks from ambient lora transmissions. SIGCOMM ’18.
- [23] A. A. Saleh and R. Valenzuela. A statistical model for indoor multi-path propagation. *IEEE Journal on selected areas in communications*, 5(2):128–137, 1987.
- [24] N. Sornin and L. Champion. Signal concentrator device, Oct. 17 2017. US Patent 9,794,095.
- [25] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota. Lora backscatter: Enabling the vision of ubiquitous connectivity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2017.
- [26] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.
- [27] A. Varshney, O. Harms, C. M. Pérez-Penichet, C. Rohner, F. Hermans, and T. Voigt. Lorea: A backscatter architecture that achieves a long communication range. Sensys’17.
- [28] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and reliable low-power backscatter networks. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 61–72. ACM, 2012.
- [29] P. ZHANG, M. Rostami, P. Hu, and D. Ganesan. Enabling practical backscatter communication for on-body sensors. In *Proceedings of the 2016 ACM SIGCOMM Conference*.