One Shot for All: Quick and Accurate Data Aggregation for LPWANs

Ningning Hou[1,2], Xianjin Xia[2], Yifeng Wang[2], Yuanqing Zheng[2]
[1]Macquarie University, Sydney, Australia
[2]The Hong Kong Polytechnic University, Hong Kong, China
ningning.hou@mq.edu.au, xianjin.xia@polyu.edu.hk,
yifeng.wang@connect.polyu.hk, yqzheng@polyu.edu.hk

***Abstract*—This paper presents our design and implementation of a fast and accurate data aggregation strategy for LoRa networks named *One-shot*. To facilitate data aggregation, One-shot assigns distinctive chirps for different LoRa nodes to encode individual data. One-shot coordinates the nodes to concurrently transmit encoded packets. Receiving concurrent transmissions, One-shot gateway examines the frequencies of superimposed chirp signals and computes application-defined aggregate functions (*e.g.*, sum, max, count, *etc.*), which give a quick overview of sensor data in a large monitoring area. One-shot develops techniques to handle a series of practical challenges involved in frequency and time synchronization of concurrent chirps. We evaluate the effectiveness of One-shot with extensive experiments. Results show that One-shot substantially outperforms state-of-the-art data aggregation methods in terms of aggregation accuracy as well as query efficiency.**

***Index Terms*—LPWAN, Aggregate Queries, Sensor Networks**

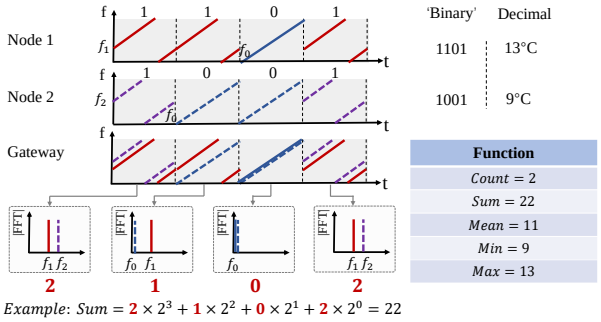


Fig. 1. An illustrative example of One-shot: Each node transmits sensing information encoded with base chirps (bit '0') and non-base chirps (bit '1') concurrently with other nodes. The gateway obtains aggregated result by counting the number of non-base chirps (bit '1') within each chirp duration.

I. Introduction

LoRa is one of the most promising Low-Power Wide Area Network (LPWAN) technologies for sensor data collection over large areas [1–9]. Benefiting from long communication ranges, LoRa is especially suitable for low-power low-rate IoT applications such as environmental monitoring [10], smart city [11, 12], smart metering [13–17], *etc*. LoRaWAN forms a star topology where LoRa end devices directly communicate with a gateway in one-hop. This network architecture eliminates the delays of multi-hop transmissions as compared to traditional hierarchical wireless sensor networks [18].

Our research targets at typical applications of using LoRaWAN for data collection and emergency response, such as disaster rescue [19] and wildfire alert [20, 21]. In those applications, quick data collections from a large number of sensors are usually required. For example, consider scenarios where toxic gases leak from a chemical plant in a city or wildfires sweep across a forest. In order to respond in a timely and effective manner, we need quick overviews of the events (*e.g.*, source of the leakage, the number of trapped people, polluted areas, *etc.*), which involves data acquisition from a large number of sensor nodes. However, due to low duty-cycles of LoRaWAN (*e.g.*, 1% in Europe [22]) and long air-time of a LoRa packet (*e.g.*, hundreds of milliseconds to several seconds), it may take several hours for a gateway to sequentially query LoRa nodes in a large monitoring area, which cannot meet the stringent time requirement of emergency response applications.

To accelerate data collection, one possible way is sparse sampling that only queries a subset of nodes [18, 23]. However, even with sparse sampling [22], it still needs to query hundreds of nodes for a large monitoring area. Latest parallel decoding technologies [24–30] support multiple nodes to communicate concurrently, which can speed up data collection to some extent. However, the maximum concurrency supported by those methods is limited to tens of LoRa nodes. There remains a large gap between the application requirements and the data collection efficiency in emergency response applications.

Recently, QuAiL [20] uses *data aggregation* method to collect statistical results of sensory data from hundreds of LoRa nodes. The rationale is that approximate overviews of all sensors would be sufficient and preferred for emergency responses rather than the exact data of individual ones. QuAiL coordinates multiple LoRa nodes to send packets simultaneously and measures signal power at a gateway to roughly estimate some statistics of sensor data (sum, mean, variance, *etc.*) which are encoded by the power levels of superimposed signals. Since many nodes transmit concurrently, QuAiL can dramatically reduce data acquisition time. However, as the power of received signals is susceptible to noise and interference, the measured power can differ substantially from the expected value, resulting in incorrect recovery of the encoded aggregation data.

In this paper, we present a ***quick and accurate data aggregation*** method for LoRa, named *One-shot*. Unlike QuAiL

that uses signal power to encode aggregation data, One-shot aggregates data in frequency domain. In particular, One-shot assigns different chirps to different nodes to encode data. Due to unique features of LoRa chirps, the superimposed signals of concurrent LoRa chirps, after de-chirp, would locate in distinctive frequency bins. We can use the number and locations of occupied frequency bins to represent aggregation data of multiple nodes. As a de-chirp operation is able to concentrate signal power of long LoRa chirps into high power peaks in corresponding frequency bins, which are resilient to noise and interference, One-shot is promising to bring accuracy and robustness to LoRa data aggregation.

One-shot develops a novel coding scheme to facilitate data aggregation for large-scale LoRa networks. For each LoRa node participating in data aggregation, One-shot employs two chirps (*i.e.*, a *base chirp* and a *non-base chirp*) to encode sensory data in a binary bit format as illustrated in Fig. 1. All nodes use the same base chirp to encode bit '0', but use different non-base chirps to encode bit '1'. Upon receiving data query commands, all participating nodes will respond simultaneously with their individual sensory data. Data aggregation is performed over the air, where the chirps of multiple nodes superimpose. A gateway examines the frequencies of superimposed signals and interprets based on different aggregation rules (*e.g.*, count, sum, mean, max, and min) to decode the aggregated data of selected nodes. As shown in Fig. 1, we can count the number of non-base chirps from superimposed signals to mimic a binary 'add' operation, which is used by One-shot to calculate aggregation functions such as count, sum and mean. Besides, as One-shot assigns different nodes with distinctive non-base chirps that differ in initial frequencies, we can separate the chirps from different nodes based on frequencies, and perform data aggregations such as max and min. Based on these basic functions, One-shot can compute a variety of data aggregation functions such as spatial and temporal distributions of sensory data in a large area via a few number of queries, which allows us to get quick and accurate information of interested areas for emergency responses.

Putting the above high-level design into practical system, however, entails two key challenges. Firstly, it is challenging to synchronize time and frequency among participating LoRa nodes in concurrent responses to One-shot queries. Due to hardware imperfection, LoRa nodes may experience frequency variations and response delays that can differ largely for different nodes. Such frequency and time deviations may alter the frequency of transmitted chirp signals, leading to erroneous data aggregation results. Second, as data aggregation may involve many LoRa nodes distributed in a large area, the signal quality of participating nodes can vary dramatically depending on communication ranges and channel conditions (*i.e.*, *near-far problem*). Due to interference among concurrent chirps, some weak chirps from far-away nodes over poor links become less likely to be detected, leading to errors in data aggregation.

One-shot develops a series of techniques to handle the above practical challenges. We find that, though the carrier frequency and response time delay vary across LoRa nodes, they can be relatively stable for a given node. Based on this observation, we can compensate for frequency deviations and time delays of different LoRa nodes to align with the frequency and time of gateway. We add small frequency gaps in between concurrent chirps to safe-guard chirps against frequency misalignment caused by noises and interference. To mitigate the impacts of near-far problem, One-shot properly allocates transmit power to the nodes participating in data aggregation. Moreover, One-shot assigns encoding chirps based on signal qualities of LoRa nodes to avoid adverse impacts of strong chirp on weak chirp detection. Eventually, it empowers a gateway to detect and decode concurrent chirps reliably for robust data aggregation.

We implement a prototype system of One-shot with software defined radios and evaluate the effectiveness of One-shot via extensive experiments. Evaluation results demonstrate that One-shot substantially outperforms the state-of-the-art methods in both data accuracy and communication efficiency. In particular, One-shot can produce higher aggregate accuracy than QuAiL in various SNR conditions and dramatically reduce data acquisition time by approximately 5.6× compared to concurrent transmission strategies.

This paper presents One-shot, whose preliminary design was presented in [31]. Here we present an extended and renewed version that includes novel designs of the data aggregation protocol (*i.e.*, query and response packet structures and data aggregation working process), theoretical analyses on the upper bound scalability of one query, new evaluations on time efficiency of data aggregation, and discussions on practical issues and possible solutions when deploying One-shot. We also include a case study to validate the effectiveness of One-shot and give readers a thorough understanding of LoRa data aggregation. These enhancements and novelties will be discussed throughout the paper.

II. Motivation

A. Target Application Scenarios

Fast and accurate data aggregation is very useful in response to disasters or rapidly evolving events. Since LoRa networks usually cover a wide space and contain a large number of LoRa nodes, it is particularly important to compute a function of distributed data acquired from nodes in real-time to make fast and accurate diagnostics. For example, in severe flooding events, we need to know some aggregated results of a large number sensor nodes deployed in the field as soon as possible. In this case, a quick and accurate estimate of max (*e.g.*, the maximum rainfall and maximum flood flow), sum (*e.g.*, the number of trapped people), and mean (*e.g.*, the average rainfall and average flood flow) can be of vital importance. Application scenarios include but are not limited to wildfire alerts and evacuation [20], toxic gases leakage assessment [32], volcano monitoring [33], asset tracking [34], and disaster rescue [19].

In addition, the data aggregation function supported by One-shot is also applicable to machine learning inference and spatial information retrieval tasks based on sensor data since the core computation functions of such tasks are linear combinations and summation of weighted sensor data [20]. As such, we see many potential applications that can benefit from the quick and accurate data aggregation service in the era of Artificial Intelligence of Things (AIoT) in the near future.

TABLE I
COMPARISON OF SCALABILITY AND NUMBER OF QUERIES FOR DATA AGGREGATION WITH PARALLEL DECODING TECHNOLOGIES.

	Feature	Scalability[1]	Num. of Queries
Choir[2][24]	Frequency	≤ 10	167
FTrack [25]	Time	3∼5	125
NScale [28]	Power	10	50
Pyramid [29]	Time	20	25
PCube [26]	Phase	30	20

[1] We compare the scalability of each method when SF is 8 and bandwidth is 250 kHz.
[2] In theory, Choir's maximum scalability is 10. But in real-world scenarios, it's scalability is 2∼3.

B. Existing Approaches

Parallel decoding mechanisms. Latest advances in LoRa networks explore unique features of LoRa to resolve packet collisions and decode concurrent packets from multiple LoRa nodes. For example, Choir [24], FTrack [25], NScale [28], AlignTrack [35], Pyramid [29] and PCube [26] classify collided symbols into the correct packets by using various physical layer features such as frequency, power, time, and phase. However, the bottleneck of applying parallel decoding to LoRa data aggregation is the limited scalability. The supported concurrency of those parallel decoding methods is far from the data aggregation requirement of a large scale LoRa network containing hundreds or thousands of LoRa nodes. Table I summarizes the scalability and the number of queries of several representative parallel decoding methods, which are used to collect data from 500 nodes.

This table shows that even with the help of parallel decoding, a gateway still needs tens or hundreds of queries to obtain the whole picture of 500 LoRa nodes in LoRaWAN. Besides, the above parallel decoding methods only work when colliding packets have sufficient time or frequency misalignment. Moreover, their scalability and accuracy degrade as SNR becomes worse. Therefore, parallel decoding methods fail to meet the tight time constraints and robustness for data aggregation in the above application scenarios.

Data aggregation. State-of-the-art data aggregation work QuAiL [20] aims to capture an approximate view of a large monitoring area by aggregating sensor data from LoRa nodes within a few seconds via a few queries. To this end, QuAiL encodes sensor data into the power of individual packets sent by LoRa nodes and derives an aggregate result from the power level of multiple concurrent packets, measured at a gateway. However, in practice, the received power at the gateway is not only affected by individual LoRa nodes but also by the dynamics of the wireless channels (*e.g.*, noise and interference). Moreover, radio signals of multiple packets can constructively or destructively interfere. As a consequence, the overall power levels of superposed signals can differ largely.

In our initial experiments, we observe that the received power of a single packet varies over time even if we fix the transmission power. We then conduct experiments with multiple LoRa transmitters and measure the power level of aggregate packets from 20 concurrent LoRa nodes. The SNR of each individual packet differs mainly because of different

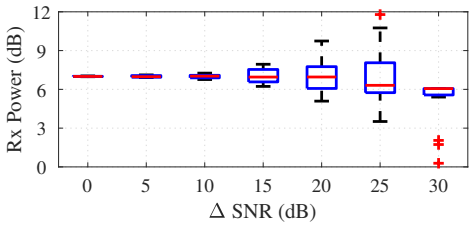


Fig. 2. The aggregated power of the state-of-the-art method is susceptible to noise and interference.

communication range from the nodes to the gateway. We use ΔSNR between packets to quantify their power variation level. As shown in Fig.2, the received power varies substantially as ΔSNR becomes larger than 10 dB, and the variation increases sharply as the ΔSNR is larger than 20 dB. This result indicates that it is challenging to acquire an accurate aggregate result by measuring the power level of colliding packets due to various influencing factors such as noise and interference, difference in communication range from nodes to the gateway, and blockage of line-of-sight paths. Our work is able to overcome the drawbacks of the state-of-the-art method and bring accuracy and robustness to LoRa data aggregation.

III. SYSTEM OVERVIEW

One-shot encodes data in binary form using CSS chirps and performs concurrent transmission to enable quick and accurate data aggregation for distributed sensors in LoRa networks. In One-shot, end devices and the gateway work together to obtain target aggregation results. End devices perform pre-processing with raw sensor data and transmit concurrently after receiving a query message from the gateway. On the other hand, the gateway coordinates concurrent transmissions of end devices by sending query messages. Once receiving responses from end devices, the gateway demodulates and decodes the concurrent transmissions and estimates the aggregation result. At a high level, the key processes of data aggregation can be represented as follows

$$h(d_1, ..., d_K) = f(\sum_{k=1}^{K} g_k(d_k)) \tag{1}$$

where $h(d_1, ..., d_K)$ is our target aggregation function, d_k is the sensor data of node k, $g_k(\cdot)$ is the pre-processing function known to all nodes, and $f(\cdot)$ is the post-processing function performed at the gateway.

As an illustrative example, let us suppose a gateway wants to construct a spatial heatmap using LoRa-enabled temperature sensors deployed in a forest, which can be very useful for heatwave tracking and wildfire monitoring. Since the temperature data of sensor nodes usually has a high degree of spatial correlation, the heatmap should be sparse and can be compressed in some domains (*e.g.*, discrete cosine transform (DCT) [20]). This gives us an opportunity to recover the heatmap by querying and aggregating n linear combinations. Therefore, we can make each node multiply its raw sensor data with the most critical top-n non-zero weights ($g_k(\cdot)$ known to the nodes) and then transmit the data concurrently with other

sensor nodes. The gateway then computes the weighted sum and applies invert DCT ($f(\cdot)$) to recover the heatmap.

As a matter of fact, with proper design of pre-processing and post-processing functions, a number of useful aggregation results can be obtained to support various applications beyond the spatial heatmap construction. For example, with One-shot, it is easy to count the number of active (or low-battery) LoRa nodes (*i.e.*, cardinality estimation) in a LoRa network by requesting each node (or low-battery node) to send one non-base chirp. The sum and the average of all sensor readings can also be efficiently aggregated. Moreover, some machine learning inference tasks over distributed sensor data can also be supported as in [20].

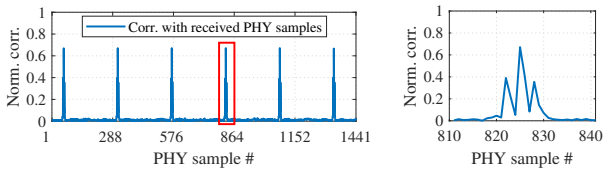
In the following, we focus on physical layer encoding and decoding at LoRa nodes and the gateway, respectively.

IV. One-Shot Design

A. Data Encoding and Decoding

One-shot takes advantages of CSS modulation by mapping data (after pre-processing) to chirps. In One-shot, packets transmitted by LoRa nodes adopt the same packet structure as traditional LoRa packets while the payload part are encoded with binary chirps. We use a base chirp to represent bit '0' and use a non-base chirp to represent bit '1'. Note that though One-shot also encodes data with chirps, the coding principle of One-shot fundamentally differs from the traditional LoRa. In traditional LoRa, chirps encode multiple bits (*i.e.*, SF bits) with different initial frequencies, which allows LoRa to achieve robust long-range communication yet at the cost of low data rates. In One-shot, one chirp only encodes one bit per chirp which would seem very limited at first glance. By aggregating concurrent nodes, One-shot can potentially increase the aggregate data rate from SF bits to approximately 2^{SF} bits per chirp duration. Moreover, as each node can send chirps at its full transmission power, the salient feature of long communication range as well as high robustness against noise and interference of traditional LoRa can be effectively retained.

On the other hand, to decode the aggregated data, a gateway needs to first detect the concurrent packets and extract chirp boundaries. Since the aggregated packets follow the traditional LoRa packet structure, a gateway can exploit standard packet detection methods [25] to detect the arrival of current packets and extract the packet chirp boundaries. In specific, a gateway detects LoRa preambles by correlating the received signal with locally generated base chirps. Fig. 3(a) shows the preamble detection result of four concurrent packets. As shown in the figure, there are periodic correlation peaks. A gateway can count the number of correlation peaks above a certain threshold (*i.e.*, ≥ 4 [36]) to detect the packets and determine the chirp boundaries. There may exist tiny chirp boundary offsets in time among multiple packets, which exhibit as multiple peaks in the FFT bins as shown in Fig. 3(b). In fact, the correlation operation is robust to small time offsets among quasi-concurrent packets. The periodic highest peaks indicate the chirp boundaries of concurrent packets. We address time and frequency synchronization issue in Section IV-C.



(a) Correlation result (b) Zoom-in result

Fig. 3. Packet detection with correlation at the gateway. (a) Correlation result of concurrent packets; (b) Zoom-in result of peaks in the red box in (a).

Since multiple nodes transmit packets at the same time and frequency, there are ensemble of chirps within one chirp modulation window. To decode the aggregated data, a gateway first performs de-chirp and FFT operations as it does in the demodulation of traditional LoRa packets. However, instead of extracting the exact FFT bin locations of concurrent chirps, One-shot only needs to count the number of non-base chirps appearing in each chirp window. To this end, we set a threshold empirically to separate non-base chirp peaks from noise after de-chirp and FFT operations. We adjust the threshold according to SNR conditions. In the context of data aggregation, as illustrated in Fig. 1, we obtain the summation of a linear combination by adding up the multiplication of the number of '1's in a chirp window with the corresponding weight of that window. We assume that a node will not transmit only '0's.

Note that depending on the data type and message length, nodes use different number of chirps to encode data to accomplish different target aggregate functions. Each node can use eight chirps (each encoding 1-bit information) to encode a byte. A packet can consist of multiple chirps to encode a different number of bytes in practice. Without loss of generality, we focus on the transmission and aggregation of data in binary format.

B. Enhancing Scalability

As introduced in the above subsection, nodes in One-shot select a non-base chirp to encode bit '1'. The number of non-base chirp candidates for a node is $N_{bin} = N - 1$, where N is the number of different initial frequencies. Since there are a large number of nodes in a network, two nodes (or more) may select the same non-base chirp to represent bit '1', resulting in collisions in the same FFT bin and leading to wrong aggregation results. Suppose there are R nodes in a network, the probability of collision (any number of nodes collide at any potential non-base chirps) can be calculated as 4 given that each node randomly selects one non-base chirp with an equal chance.

$$\begin{aligned} P_{co} &= 1 - \frac{N_{bin}}{N_{bin}} \times \frac{N_{bin}-1}{N_{bin}} \times \frac{N_{bin}-2}{N_{bin}} \cdots \frac{N_{bin}-R+1}{N_{bin}} \\ &= 1 - \frac{N_{bin}!}{(N_{bin}-R)! \times {N_{bin}}^R} \end{aligned} \tag{2}$$

For example, if there are $R = 10$ nodes in a network, and the SF is 8 ($N_{bin} = 2^{SF} - 1$), the probability of collision is $P_{co} = 16.36\%$. As the number of nodes increases to 20, P_{co} becomes 53.50%. As the number of nodes R further increases,

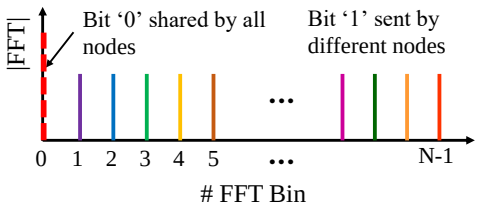


Fig. 4. One-shot assigns different initial frequencies for different nodes to encode bit '1', and all nodes use the base chirp (bin 0) to represent bit '0'.

P_{co} approaches 1 rapidly. If R is larger than N_{bin}, it is for sure that two nodes have to encode '1' with the same non-base chirp.

Intuitively, we can decrease the probability of co-location collisions by adopting a large SF to increase the number of available non-base chirps to support more nodes. A wider bandwidth (BW) is also helpful for increasing the frequency gap between two adjacent non-base chirps to mitigate the interference from each other. However, even with the optimal SF and BW configurations, some nodes may still select the same non-base chirp.

To avoid such co-location collisions and enhance the scalability, One-shot assigns different initial frequencies to different nodes. Fig. 4 illustrates the high level idea of frequency assignment scheme. FFT peaks with different colors represent bit '1' sent by different nodes. This simple yet effective method can avoid collisions of chirps in frequency domain and decrease aggregation estimation error. In practice, One-shot can pre-configure an initial frequency for each node when the node joins the network, or dynamically assign different initial frequencies to nodes by leveraging a collision-free hash function that maps each node to a distinct initial frequency. Ideally, One-shot can support a maximum number of $N_{node(max)} = 2^{12} - 1$ ($SF = 12$) nodes in one query if all nodes are well-synchronized and in good channel conditions. Since multiple base chirps may appear in the first FFT bin, the interference to non-base chirps can be strong. To mitigate the impact on counting non-base chirps, we set 6 guard bins between the base and non-base chirps.

In case the total number of nodes R is larger than the maximum available non-base chirps, One-shot splits all nodes into several batches and queries the nodes batch-by-batch. The batch index of a node can be pre-configured. It then combines aggregation estimation results of all batches to derive the final aggregation result.

C. Addressing Practical Issues and Our Solutions

The performance of One-shot can be affected by various practical factors. For example, commodity devices are imperfect and real-world wireless channels are subject to noise and interference, limiting the maximum number of concurrent transmissions. In the following, we discuss the practical issues and our solutions to mitigate their impact and optimize the scalability and accuracy of data aggregation.

Collision of non-base chirps. Ideally, if we assign different non-base chirps with different initial frequencies to LoRa nodes, their corresponding peaks should appear in different

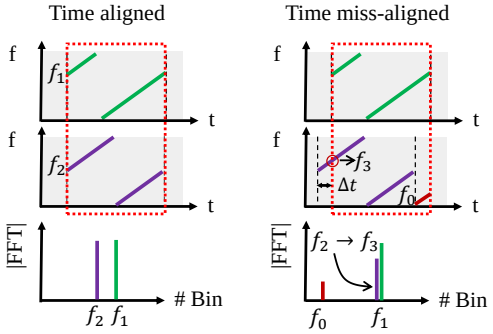


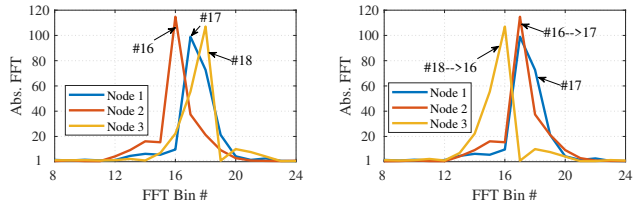
Fig. 5. Impact of time misalignment on decoding.

FFT bins, *i.e.*, no collision of non-based chirps in theory. This design requires all devices to transmit different non-base chirps at the same time with the same carrier frequency (*i.e.*, synchronization in both time and frequency). In practice, however, due to a variety of practical factors, it is challenging to guarantee tight synchronization among a large number of LoRa nodes in time and frequency.

• Time misalignment. As shown in Fig. 5, suppose there are two nodes sending packets concurrently. If non-base chirps transmitted by the two nodes are well-aligned in the demodulation window (dashed red box), One-shot decoder can obtain two peaks with correct initial frequencies. In contrast, if there is a time shift (Δt) between these two chirps (the demodulation window is misaligned with the second chirp), the FFT result of the second chirp will shift to a different FFT bin, *i.e.*, from f_2 to f_3. Besides, a portion of the adjacent chirp (the red base chirp, with initial frequency of f_0) leaks into the current demodulation window, causing interference with chirps from other nodes and decoding errors.

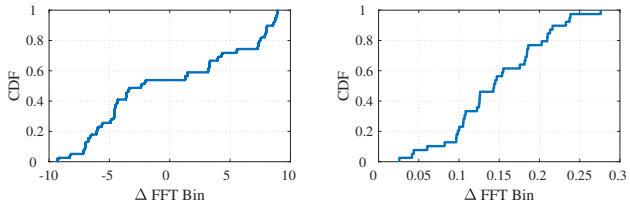
There are two factors that can cause time misalignment. The first one is *response time variance of different nodes*. We measure the response time distribution of 40 LoRa nodes and observe that the response time varies from 584μs to 592μs with a standard deviation of 3.47 μs. The maximum response time difference is 8 μs, which is 1 FFT bin when BW is 125 kHz. That means if nodes immediately respond to a query message from a gateway, their concurrent transmissions can be aligned in time. However, when BW is larger than 125 kHz, the response time difference may lead to more than one frequency bin shift. The second impact factor is *propagation delay*. Due to the long communication range, the distances between nodes and a gateway can vary a lot in LoRa networks. Therefore, two nodes with a considerable distance in between can not receive the query message at the same time, let alone synchronize their response messages at the gateway. A 8 μs time delay corresponds to 1 FFT bin shift ($BW = 125$ kHz) and a propagation distance of 2.4 km. For example, the FFT peak may shift 5 FFT bins from the assigned bin when the distance between two nodes is 6 km (a round trip of 12 km).

One-shot compensates for the difference in propagation distance by adding different response delays according to the distance between LoRa nodes to the gateway. The geographical locations of nodes and the gateway can be pre-configured into the devices during node deployment or obtained via



(a) (b)

Fig. 6. (a) Non-base chirps without CFOs located at bin 16, 17, and 18, separately. (b) Non-base chirps with CFOs shift to wrong bins and cause collisions.



(a) (b)

Fig. 7. (a) CFO distribution of 40 nodes (where Δ FFT Bin is the normalized CFOs to bins as BW=250 kHz and SF=8). (b) Frequency-drift distribution of one single node.

localization services (*e.g.*, GPS, LoRa localization [37, 38]). As One-shot only needs a km-level localization accuracy (corresponding to 1 FFT bin), we assume that such localization services are available for LoRa nodes. In our implementation, the geo-location of nodes and their corresponding delays are calculated and hard coded before deployment. In this way, after receiving a query message from a gateway, nodes located closer to the gateway will wait for the distant nodes, so that their concurrent transmissions will be aligned in time.

- Frequency misalignment. Frequency misalignment between nodes and a gateway results in shift of initial frequency at the gateway. Such frequency misalignment can cause severe interference between chirps with similar initial frequencies. Fig. 6 shows an example. Three nodes are pre-configured with non-base chirps corresponding to bin 16, 17, and 18, respectively. Without CFOs between nodes, the peaks of non-base chirps locate at the correct bins in Fig. 6(a). In contrast, when there are CFOs among these nodes, the chirps shift to wrong bins and interfere with each other. For example, the non-base chirp of node 2 shifts from bin 16 to bin 17 and interferes with the signal from node 1, causing decoding errors.

LoRa nodes are low-cost devices and may have carrier frequency offsets (CFOs) among nodes due to hardware imperfections (clock drifts and variations). We evaluate the CFOs among 40 commodity LoRa nodes and plot the result in Fig. 7. We observe that the carrier frequency varies among different nodes ($\pm$ 10 FFT bins) (Fig. 7(a)). The clock of a single node also drifts (Fig. 7(b)), but remains relatively stable over time (maximum of $\Delta FFT_{bin} = 0.3$).

Based on this finding, One-shot calibrates each node and configures the central frequency of each node before data aggregation. In particular, One-shot estimates the CFO between each node and the gateway by examining the multiplication result of SFD down-chirp and base up-chirp from the same packet as in the literature [26, 39–41]. Then, One-shot compensates for the frequency offset by adding the estimated CFO to the central frequency of each node. The CFO estimation and calibration can be performed regularly after routine message exchanges between each node and the gateway (*e.g.*, beacon, ACK) before the data aggregation operation. The nodes with excessively large CFOs can be identified and calibrated immediately by examining the spectrum of concurrent transmissions, where the chirps with large CFOs ($\pm$ 10 FFT bins) will substantially deviate from the correct central frequency.

To further mitigate the impact of time and frequency misalignment, One-shot sets guard bins between the selected frequency bins to tolerate potential time and frequency shifts and interference. Specifically, we choose non-base chirps corresponding to every M bin as candidate bit '1', where M is the number of empty bins between two available non-base chirps. One-shot can adopt a larger M to mitigate the inter-packet interference and improve the robustness of decoding in bad channel conditions. Yet, a larger M lowers the scalability of data aggregation. Suppose the maximum number of nodes in a batch is $N_{b_{max}}$. In theory, $N_{b_{max}}$ can be calculated as the following equation. Note that we set 6 guard bins between the base and non-base chirps.

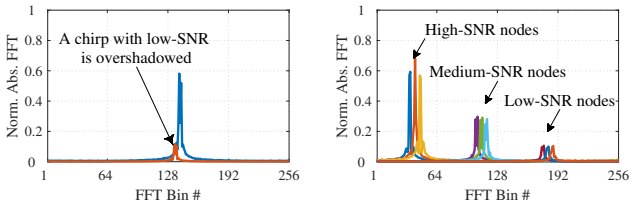
$$N_{b_{max}} = \lfloor (\frac{2^{SF} - 6}{M}) \rfloor \tag{3}$$

In practice, One-shot needs to strike a balance between scalability and robustness by tuning M. We conduct an experiment in Section V-B2 to evaluate the impact of M. We can also observe from Eq. 3 that SF also influences the maximum number of nodes in a batch. A larger SF means more available non-base chirp candidates. However, a larger SF also makes the frequency gap between two candidates smaller. We evaluate the impact of SF in Section V-B2. In addition, One-shot zero-pads the received signals to get a fine-grained location estimation of peaks to facilitate counting of non-base chirps.

Note that Eq. 3 does not imply that One-Shot can only support data aggregation with $N_{b_{max}}$ nodes. For a large network with more than $N_{b_{max}}$ nodes, One-shot first divides all nodes into different batches and then queries them batch-by-batch. It can finally obtain the aggregation results by merging the results of different batches.

Near-far problem. As concurrent nodes concentrate their transmission power in their assigned FFT bins, we expect strong robustness against background noise and interference. Yet, in practice, we notice that the signal strength from a nearby node can be much stronger than that from a remote node. As shown in Fig. 8(a), overshadowed by a high peak, an adjacent weak peak can be easily missed, leading to an inaccurate aggregate estimate. The problem becomes severer in a typical below-the-noise-floor condition over a long range communication in LoRa networks.

To address this problem, we assign neighboring FFT bins to nodes with similar SNR levels. Fig. 8(b) illustrates intuition of



(a) (b)

Fig. 8. (a) The side lobe of a strong chirp may overshadow the main lobe of a weak chirp. (b) Assign nodes with similar SNR level to neighbouring FFT bins to facilitate weak peak detection.

this idea. Chirps from high-SNR nodes are grouped together while chirps from low-SNR nodes are assigned with FFT bins far away from the high-SNR nodes. One-shot pre-configures a non-base chirp for each node when a node joins the network. In our design, a node initiates its join request with medium transmission power. If a node cannot receive an ACK from the gateway, the node re-transmits the join request with the maximum power. A gateway determines the initial frequency of a non-base chirp according to the received signal power of the join request and the signal power of existing nodes in the network. In this way, nodes with asymmetric SNRs are separated into different ranges of FFT bins. Lower SNR devices will be assigned to FFT bins far from those of higher SNR devices. The side lobe of a strong peak will not interfere with the main lobe of a weak peak far from it, as the side lobe's power becomes weaker when it becomes farther from the main lobe. This non-base chirp assignment method facilitates weak peak detection. In a larger network, One-shot can assign different sets of nodes to different batches depending on their signal strengths, *i.e.*, nodes that have a similar SNR are grouped into the same batch to further minimizing the near-far problem.

Power adaptation. One-shot adopts power adaption to reduce the power consumption of LoRa nodes and account for channel variations in the process of data aggregation. Intuitively, a nearby node can use lower transmission power to save energy. In contrast, a remote node has to send a packet with adequate power to ensure correct reception and decoding at the gateway. Besides, a node needs to adjust its transmission power in case of channel variations to guarantee a good performance of data aggregation estimation.

We leverage the channel reciprocity between a node and a gateway to adjust the transmission power. In particular, a node changes its transmission power according to the query packet received from a gateway. Intuitively, if the signal strength of the query packet is high, according to the channel reciprocity, a node can conservatively reduce its transmission power to save some energy without affecting communication performance. In data aggregation process, a node self-adjusts its transmission power according to the power of the received query message. Note that One-shot pre-configures neighboring FFT bins to nodes with similar SNR levels. Though a node can adjust its transmission power dynamically, it would be better to change its transmission power conservatively instead of drastically. In practice, we can set a changing range for a node to combat channel dynamics.

Query pkt	Preamble [6]	Sync. Words [2]	SFD [2.25]	Batch ID [2]	Data Type [2]
Response pkt	Preamble [6]	Sync. Words [2]	SFD [2.25]	Batch ID [2]	Sensor Data [Adaptable]

Structure	Functionality
Preamble	Indicate presence of a query message, frame timing detection, etc.
Sync. Words	Encode network ID for authentication
SFD	Signal start of the actual frame
Batch ID	Encode the batch ID of a node
Data Type	Indicate sensory data type, e.g., temperature/humidity
Sensor Data	Sensory data in binary form

Fig. 9. Packet structure of query and response messages in One-shot.

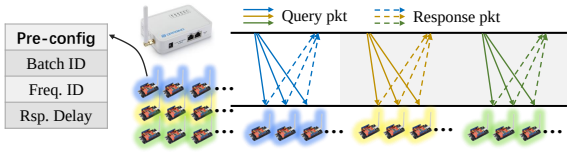


Fig. 10. One-shot data aggregation working process.

Query and response message. Fig. 9 shows the packet structure of query and response messages. Both the query message and response message have a preamble (6 base chirps) and mandatory sync words and SFD, which are the same as the current LoRa packet. After SFD, both of them use two chirps to encode the batch ID of a node. The query message uses two chirps after the batch ID to indicate what kind of sensory data it is querying. For the response message, the following part is the sensor data. The length of sensor data varies according to queries of different types of sensing data. Note that only the sensory data is modulated using base chirps and non-base chirps. The other parts of the query and response packets are modulated in CSS.

D. One-shot working process

Note that each LoRa node in the network is pre-configured with batch ID, non-base chirp frequency ID, and response delay according to their location. Such pre-configuration is a one-time setup. A node also compensates for the carrier frequency offset and determines its transmission power according to the feedback from a gateway in the joining procedure to the network. Since the oscillator and the location of a node are relatively stable, the frequency compensation measurement and power adjustment are infrequent.

Fig. 10 summarizes the data aggregation process of One-shot. Firstly, a gateway broadcasts its query, which contains the target batch ID and sensory data type. Upon receiving the query packet, nodes with the target batch ID respond to the query using the response packet shown in Fig. 9. If the number of nodes in the network is smaller than the maximum number of nodes in a batch ($N_{b_{max}}$), One-shot can get the aggregate results through one query. While for a network with a large number of nodes, One-shot divides LoRa nodes into batches and then queries them one by one to obtain the aggregation results.

V. Implementation and Evaluation

A. Implementation

We implement a prototype system of One-shot with Software Defined Radio (SDR) devices (*i.e.*, USRP N210s with WBX daughterboards). We use USRPs as end nodes to send encoded One-shot packets at different locations. Limited by the number of available USRPs, we use each USRP to emulate the concurrent responses of multiple nodes by sending multiple One-shot packets encoding different sensory data.

We use another USRP as a One-shot gateway. The gateway broadcasts query messages to end nodes and receives the concurrent response packets from end nodes. The received signals of concurrent responses are first sent to a workstation and then processed in MATLAB to decode aggregation results.

B. Evaluation

1) Methodology: **Experiment setup.** We collect data traces with real channel measurements from USRPs. The USRPs are set with a slightly different central frequency to emulate the hardware imperfections of COTS LoRa nodes. We configure the central frequency offset among nodes evenly distributed between [-10, 10] kHz. The distribution of CFO is similar to that in Fig. 7. The total number of traces is over 4000. We add additive white Gaussian noise at the gateway to evaluate the system performance under different channel conditions. For each set of evaluation, we repeat the experiment 50 times and report the average result. Query packets and response packets are structured as shown in Fig. 9. We set the sensor data length of the response message to 50 chirps. As such, One-shot nodes transmit 50 bits per packet since one chirp can encode one bit. Unless specified otherwise, we configure the default central frequency, spreading factor, and bandwidth of the signal to 915 MHz, 8, and 250 kHz, respectively. The default guard bin M is 6.

Evaluation metrics. For simplicity, we take the calculation of the *sum* of sensing data as an example to illustrate the performance of One-shot. In real-world applications, users can measure the performance of One-shot in other aggregate functions (count, mean, and spatial distribution, *etc.*). We evaluate the overall performance of One-shot with three key metrics:

Aggregation error. We define this metric as the deviation of estimated sum compared to the ground truth of the sum of sensing data in one query. Here n is the total number of queried nodes in one query.

$$Error = \frac{|Agg._{est.} - \sum_{i=1}^{n} d_i|}{\sum_{i=1}^{n} d_i} \quad (4)$$

For QuAiL, we encode different sensory data with different power levels of packets and add all packets up at the gateway. We measure the power ($P_{agg.}$) of the aggregated data and then we perform linear regression to learn the mapping between aggregation results sum ($sum_{agg.}$) and total power ($P_{agg.}$) at the gateway. With the pre-learned mapping function, we can estimate the aggregation results ($sum_{est.}$) given a measured aggregation power ($P_{measure.}$). We calculate the aggregation error rate of QuAiL as

$$Error = \frac{sum_{est.} - sum_{truth}}{sum_{truth}}$$

which has the same rationale as the aggregation error metric (*i.e.*, Function 4) for One-shot.

Number of queries. We use the number of queries needed to extract a target aggregate function of the whole network to evaluate the time efficiency of data aggregation. For convenience, we assume the network scale is 500. Each node needs to be queried at least once. A gateway only accepts a query result with an aggregate error of less than 10%.

Goodput. The goodput of one query is the amount of sensing data in bits that are successfully received by the gateway per second, and it can be used to evaluate the data aggregation efficiency. The goodput in this context refers to the collective goodput of all nodes in one query, rather than that of an individual node. As for QuAiL, since it cannot obtain an accurate data rate for each node, we use the aggregation error of QuAiL to represent the average data error rate of each node. We assume each QuAiL node can leverage different power levels to encode 8-bit data. The goodput of QuAiL is calculated as

$$goodput = \frac{Num_{node} * 8 * (1 - Error)}{T_{trans}}$$

where T_{trans} is the transmission time of QuAiL packet.

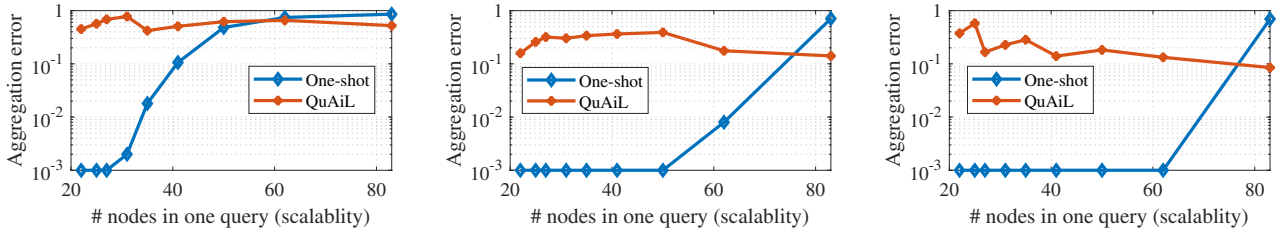
Benchmark. We compare One-shot with the state-of-the-art method QuAiL, sequential query, and two concurrent transmission methods, *i.e.*, FTrack and Choir.

2) Performance: **Robustness and accuracy.** We first evaluate One-shot's accuracy in retrieving an aggregation result that calculates the sum of data from all nodes. We compare One-shot with QuAiL under three SNR conditions (*i.e.*, -10 dB, 0 dB, 10 dB) and repeat each experiment 50 times.

If all data bits are correctly received, we mark the error as $\leq 10^{-3}$ in the figures. In the experiments, we control the number of nodes participating in data aggregation. A smaller number of queried nodes in one query can mitigate inter-node interference and improve aggregation accuracy, but at the cost of longer data collection time.

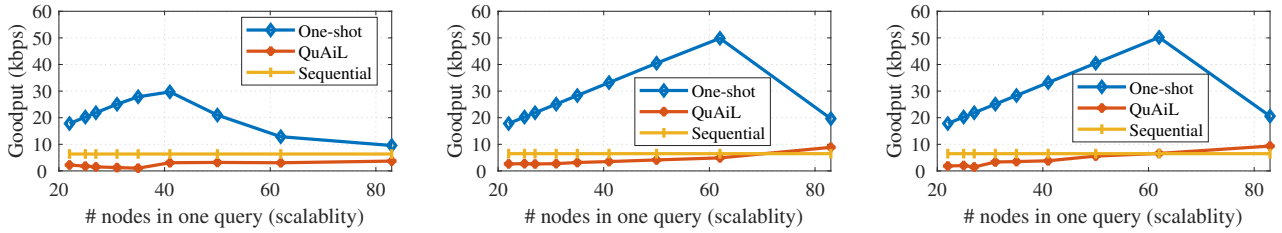
The results in different SNR conditions are shown in Fig. 11(a-c). We see that the overall errors of One-shot are much lower than QuAiL in all three SNR conditions when the number of concurrent responses is smaller than 60. As the SNR becomes higher, the accuracy of both One-shot and QuAiL improve. For instance, One-shot yields acceptable aggregate accuracy (*i.e.*, error $< 10\%$) when SNR is -10 dB. As SNR increases to 0 dB, the error further decreases. In comparison, the aggregate errors of QuAiL remain consistently higher than 10%. This experiment demonstrates that One-shot is more accurate and robust than QuAiL.

Aggregation efficiency (Goodput). In this experiment, we compare the aggregation efficiency of One-shot in terms of goodput against QuAiL and sequential queries in different settings. For QuAiL, each node can encode at most 8 bits per packet by varying transmission power levels of a packet.



(a) SNR= -10 dB (b) SNR= 0 dB (c) SNR= 10 dB

Fig. 11. Aggregation error under different SNR conditions.



(a) SNR= -10 dB (b) SNR= 0 dB (c) SNR= 10 dB

Fig. 12. Goodput comparison of One-shot and QuAiL under different SNR conditions.

For sequential queries, we report the maximum goodput of a standard LoRa node as its goodput.

Fig. 12(a-c) present the results of goodput in three SNR conditions. As expected, One-shot produces higher goodput than QuAiL and sequential query. One-shot achieves nearly 10 $\times$ improvement in goodput compared with QuAiL even when the SNR is -10 dB. The gain mainly comes from using CSS chirps to encode data since the chirps can concentrate the energy and deliver data with high robustness. In contrast, QuAiL heavily relies on linear combinations of power from multiple nodes, which can be adversely influenced by noise and interference in real-world deployment. We see slight improvement of goodput for QuAiL as the number of queried nodes increases. That is because as more nodes transmit together, the impact of power difference among nodes becomes less prominent. Yet, the overall aggregate power measurement is still susceptible to background noise and interference.

We can also observe from Fig. 12(a-c) that the goodput of three schemes improve as SNR becomes higher. The goodput of One-shot increases with the number of concurrent responses. The number reaches 49.84 kbps and 50.2 kbps when the SNRs are 0 dB and 10 dB, respectively. One-shot yields the highest goodput when 60 nodes participate in a query concurrently. As the number of nodes further increases, the goodput of One-shot starts to decrease because of inter-node interference. To ensure reliable data aggregation, we can estimate the number of nodes in a network and coordinate the nodes accordingly. For example, we may assign nodes to different batches each consisting of 40-60 nodes and aggregate data batch-by-batch.

Query latency. This experiment compares the number of queries required to obtain an aggregate result (*i.e.*, sum) from 500 nodes under different SNR conditions. We use the number of queries instead of the total amount of query time to characterize latency because different applications may have different query intervals. We compare One-shot with QuAiL and two concurrent transmission methods (*i.e.*, FTrack [25] and Choir [24]). For these two concurrent methods, we group 10 nodes in a batch. Note that our goal is to obtain an aggregate result instead of individual information of each node. The gateway accepts a query result only when the aggregate error is less than 10%. We set the scalability of FTrack and Choir under different SNRs according to evaluation results in [24, 25].

As shown in Fig. 13, One-shot needs a minimum number of 13 and 9 queries when SNR is -10 dB and 0 dB. By contrast, QuAiL can hardly yield an accurate aggregation result when SNR is -10 dB. The aggregation error of QuAiL is larger than 10% in such condition since the power of received signals is susceptible to noise and interference. QuAiL can support data aggregation of hundreds of nodes in each round of query when SNR is 10 dB and thereby reduces the total number of queries to 5 rounds. Though FTrack and Choir support concurrent transmissions of multiple nodes, they still need over 100 rounds of queries when SNR$\geq$ 0 dB. More queries are required as SNR is -10 dB. The results prove the effectiveness of using data aggregation on reducing data collection time.

Impact of different bandwidths. In this experiment, we fix SF to 8 and vary the bandwidth of LoRa chirps to evaluate the impacts of bandwidth on data aggregation. Specifically, we evaluate One-shot with BW=125 kHz, 250 kHz, and 500 kHz, respectively. We set SNR to 0 dB and measure the aggregation error and goodput of One-shot with different number of concurrent nodes.

We can observe from Fig. 14 that wider bandwidth generally produces lower aggregation error under the same scalability. The non-base chirp frequency gap between two nodes is larger when a wider bandwidth is adopted. For example, the aggregation error is less than 0.8% even when 60 nodes respond to a query simultaneously. Fig. 15 reports that wider bandwidth brings higher goodput. For instance, the goodput

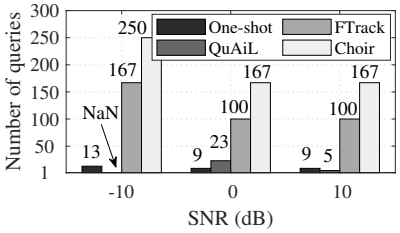


Fig. 13. Data aggregation latency.

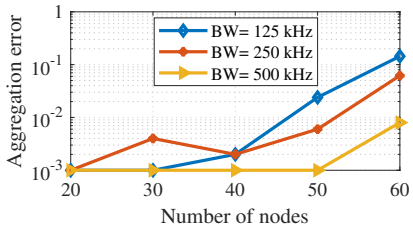


Fig. 14. Impact of BW on decoding error.

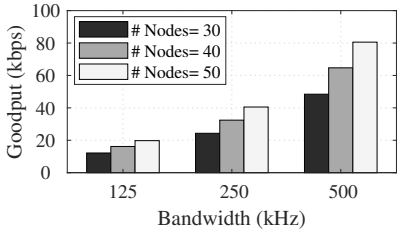


Fig. 15. Impact of BW on goodput.

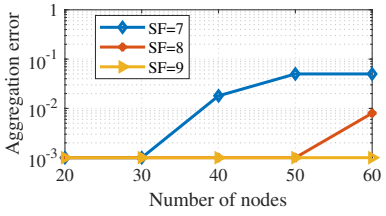


Fig. 16. Impact of SF on decoding error.

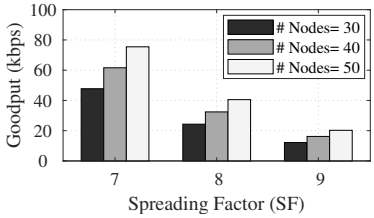


Fig. 17. Impact of SF on goodput.

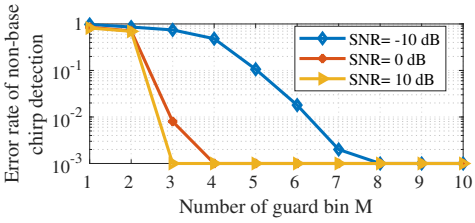


Fig. 18. Impact of the number of guard bin M.

is 80.56 kbps when 50 nodes transmit simultaneously when the bandwidth is 500 kHz. It is about twice the goodput of bandwidth 250 kHz (40.52 kbps) and four times that of 125 kHz (19.77 kbps). The results are reasonable since a packet in wide bandwidth has a short transmission time (*i.e.*, high PHY bit rate).

Impact of different spreading factors. In this experiment, we fix the bandwidth and vary the Spreading Factor (SF) of LoRa chirps. For demonstration, we set the SF of LoRa chirps to 7, 8, and 9. In this experiment, the BW is 250 kHz and the SNR is 0 dB. Other experiment configurations are the same as in the above experiment.

Fig. 16 evaluates the impact of SF on aggregation error of One-shot. Similar to standard LoRa, One-shot has lower aggregation error with larger SF. Chirps with larger SF are more resilient to noise and interference. For example, even when 60 nodes respond to a query concurrently, the decoding error is almost zero. However, the goodput of One-shot decreases as the SF increases. In Fig. 17, the goodput of $SF = 7$ (75.45 kbps) with 50 concurrent nodes is about twice the goodput of $SF = 8$ (40.52 kbps) and four times that of $SF = 9$ (20.26 kbps). The reason is that chirps with a high SF would have a long chirp duration, which increases packet air-time and reduces the overall goodput.

Impact of the number of guard bins (M). In this experiment, we evaluate the error rate of non-base chirp detection in each payload window under different settings of guard bins. We vary the number of guard bins from 1 to 10 at a step of 1 FFT bin. We calculate the ratio of windows with incorrect chirp detection to the total number of payload windows as error rate of non-base chirp detection.

We present the results in Fig. 18. As expected, a large M brings low error rates for chirp detection under the same SNR conditions. The error rate becomes lower as SNR increases. It means that a large M is required for low SNRs in order to achieve good performance on chirp detection. For example, when SNR is -10 dB, 8 guard bins are required to achieve zero error rate of chirp detection. In contrast, only 3∼4 guard bins are needed when SNR≥0 dB. One-shot can choose different M to adapt to different SNR scenarios.

TABLE II
ENERGY CONSUMPTION OF DIFFERENT WORKING MODES.

Mode	Rx	Tx	Idle	Sleep	Standby	Synthesizer
Power	10.8 mA	120 mA	1.5 μA	0.2 μA	1.6 mA	5.8 mA

Energy consumption. It is important to note that our measurements reflect a node's total energy consumption throughout the data aggregation process, rather than focusing solely on the transmission of response packets. One-shot works in a Class B-like mode: a node periodically wakes up, listens for incoming queries, transmits responses, and then goes back to idle. In Class B, a node spends most of the time ($> 99\%$) in Idle mode (with only the RC circuit oscillator turned on). If a query message is detected, the node will keep on receiving the message. Otherwise, the node is switched off until the next receive period if it cannot detect a query message. We assume each of the wake-up slots has query messages. We calculate the corresponding time of each state of a node according to our query and response messages design (with default parameter settings) represented in Fig. 9. Table. II shows the power consumption of a commodity node in different modes. For simplicity, we calculate the overall energy consumption of a packet with only Rx, Tx, and Idle modes. We measure the energy consumption of a packet and the overall lifetime of a single LoRa node under different transmission intervals (*i.e.*, 10 min, 1 h, and 24 h, respectively). We assume the end node is powered by an AA battery (*i.e.*, 1.5 V and 3000 mAh).

Table. III lists the results under different query intervals. We can observe that the transmission time of Class B node using standard LoRa modulation has approximately $4\times$ smaller transmission time and thus saves energy consumption in the transmission period. However, since the transmission time represents a tiny fraction of the query interval, the overall energy consumption of One-shot and standard Class B is

TABLE III
ENERGY CONSUMPTION.

	Int.	Tx.(ms)	L.(ms)	E/pkt(mJ)	Life(mon.)
One-shot	10min	62.25	256	11.3	0.1
Class B	10min	15.38	256	5.5	0.2
One-shot	1h	62.25	256	15.8	0.4
Class B	1h	15.38	256	10.2	0.6
One-shot	24h	62.25	256	139.8	1.1
Class B	24h	15.38	256	134.2	1.1

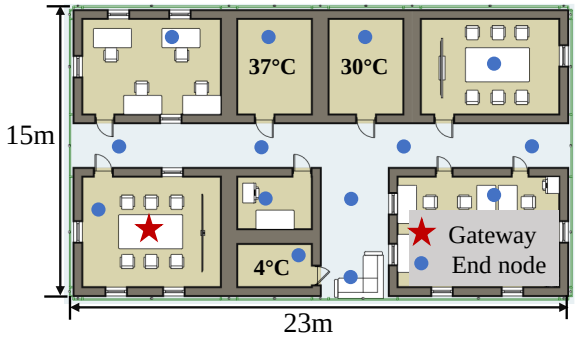


Fig. 19. Floor plan of labs and offices in the department of Applied Biology and Chemical Technology (ABCT).

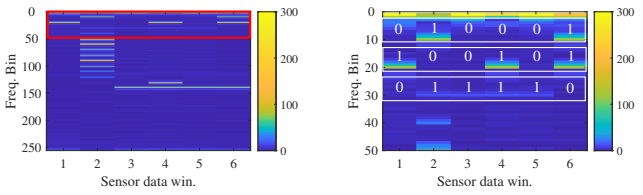
almost the same as the transmission interval becomes larger. The reporting interval has a greater impact on the lifetime of nodes.

We remark that standard LoRa and One-shot are designed for different applications. While standard LoRa modulation can enable a single node to transmit the same amount of data bits with lower energy consumption, this power conservation mode often results in longer data aggregation times that render it unsuitable for time-critical applications. In contrast, One-shot is designed to facilitate rapid data aggregation, making it capable of responding promptly to emergencies. In such cases, additional power consumption overheads are acceptable tradeoffs for quick data aggregation. In real IoT applications, a node can shift between standard LoRa transmission and One-shot to adapt to different applications and balance energy efficiency and time efficiency.

VI. CASE STUDY

In this section, we conduct a case study to verify the effectiveness of One-shot in querying the temperature data. Fig. 19 shows the floor plan of our evaluation environment. This floor spans 15m×23m. Besides offices and meeting rooms, there are three laboratories for constant temperature and humidity experiments on this floor. The temperature of these three rooms needs to keep 37°C, 30°C, and 4°C, respectively. Researchers in the Department of Applied Biology and Chemical Technology (ABCT) sometimes need to conduct experiments in a constant temperature and humidity environment.

Similar to the implementation in Section V, we use a USRP N210 as a gateway and other USRPs as LoRa end nodes. The gateway is denoted with a red pentagram, and end nodes are denoted with blue dots in Fig. 19. We also put thermometers near end devices to measure temperatures in rooms and corridors. The measured temperature is encoded in binary form, same as the example in Fig. 1. The sensor



(a) (b)

Fig. 20. (a) Spectrum of demodulated sensor data. (b) Zoom-in result of signals in red rectangle.

data length is set to 6. Note that we only encode the integral part of the temperature. Therefore, a length of 6 chirps is enough to encode the temperature. Since we have limited number of USRPs (*i.e.*, 8 USRPs), we divide the end nodes into two groups and query these nodes twice. Note that all end devices share the same batch ID. We collect the data trace of two responses and synthesis them together to simulate one concurrent response from 14 end nodes. The other parts of query and response packets are formed according to the packet structure in Fig. 9. We set M = 10 as the frequency guard bin. The central frequency, spreading factor, and bandwidth of the packets are set to 915 MHz, 8, and 250 kHz, respectively.

Fig. 20(a) plots the spectrum of demodulated aggregation data. The X-axis represents 6 sensor data windows, and the Y-axis represents frequency bins. End nodes are assigned different frequencies to transmit non-base chirp signals. Fig. 20 is color coded. A lighter color indicates a higher signal power. A horizontal strip corresponds to the location of non-base chirps sent by a specific end node. We can observe from Fig. 20(a) that there are multiple horizontal strips that contain non-base chirp signals. Fig. 20(b) shows the zoomed in results of 5 end nodes. We can observe that the sensor data of the first node is '010001', which is 17°C. The second sensor data is '100101', corresponding to the temperature in 37°C lab. Similarly, we can decode the third strip as '011110', corresponding to the 30°C lab. The decoding results of other strips also match the ground truth. We can obtain many aggregation functions (*e.g.*, mean, max, and average) from the spectrum of a demodulated graph. This study demonstrates that we can obtain the aggregate result with a single shot of query.

In this study, we only query sensor data from 14 end nodes. The non-base chirps only show up in the upper part of the frequency bins in Fig. 20(a). In practice, as evaluated in Section V-B, One-shot can support more end nodes in one query. This case study focuses on temperature sensor data aggregation on one floor; we envision One-shot supporting various future applications.

VII. DISCUSSION AND FUTURE WORK

One-shot is a prototype design for sensory data aggregation for LoRaWANs. We tackle a series of challenges to implement One-shot to support quick and accurate data aggregation. However, there is room for further work and improvement. We discuss two points here.

Data aggregation in ultra-low SNR scenarios. In our work, we test the performance of One-shot when the SNR is −10 dB. Results show that One-shot works well and can

support up to 40 nodes in a batch. However, when SNR becomes lower (*e.g.*, -20 dB), One-shot aggregation error increases sharply. Noise and inter-symbol interference make it hard to detect weak peaks. To support data aggregation in ultra-low SNR scenarios, One-shot can borrow ideas from MALoRa [36]. MALoRa combines multiple preamble chirps to detect the presence of weak packets. End nodes in One-shot can also use multiple consecutive identical non-base chirps to represent one bit. A receiver can enlarge the demodulation window to concentrate the energy of multiple chirps. Applying an error correction code is another possible way to improve One-shot. We leave them for future work.

Multi-gateway scenario. Currently, One-shot mainly focuses on single-gateway scenarios, where nodes are associated with a specific gateway. In practical settings, however, other gateways in the network may also send query requests. To address this issue, One-shot can be configured on a query basis. Since different gateways may have different central frequencies and distances to the node, we can write down parameters associated with each gateway like what we do for a single gateway, and make a looking-up table beforehand. With the looking-up table, end nodes can adjust parameters according to different initiating query gateways. In practice, a LoRa node only associates with several gateways.

Security and privacy. In the current prototype, end nodes transmit data without security or privacy protection. If a malicious adversary overhears the network communication, it may infer useful sensory information. We plan to explore whether we can integrate advances in lightweight security algorithms for low-power IoT devices [42] into One-shot.

VIII. Related Work

There has been much past work [43] on data aggregation in wireless sensor networks (WSNs). In WSNs, various methodologies, such as the tree-based approach [44] and the cluster-based approach [45], are used to collect and combine sensor data in a region of interest. Machine learning based protocol design [46] are also used in data aggregation in WSNs. Since LoRa nodes communicate with a gateway through one-hop uplinks, complex routing protocols for data aggregation in WSNs are not suitable for LoRaWAN. Our work is most related to data aggregation methods for LoRa networks. In QuAiL [20], all nodes encode their data with signal power and transmit together. Then, QuAiL measures the aggregated signal power of colliding packets at a gateway and estimate the aggregate result with the aggregated signal power. Compared with the sequential query method, QuAiL can substantially reduce the overall data transmission time by supporting concurrent transmissions of LoRa nodes. In practice, however, the measured signal power is subject to background noise and interference in the wireless channel. As a result, the estimate result can be far from the ground truth. Same as QuAiL, our work One-shot coordinates LoRa nodes to send packets concurrently and estimate an aggregate result from the colliding signals. Unlike QuAiL which encodes data with signal power, One-shot encodes data with non-base chirps. As such, One-shot effectively inherits the robustness of CSS modulation against noise and interference.

Concurrent transmission methods aim to resolve packet collisions and decode concurrent packets from multiple LoRa nodes [47, 48]. For example, Choir [24] measures the frequency offsets of chirps and resolves LoRa packet collisions. FTrack [25] leverages the misalignment of LoRa chirps to disentangle collided packets in the time domain. mLoRa [49] detects the time offset between concurrent packets based on preamble correlation results. CoLoRa [50] classifies LoRa symbols to their corresponding LoRa packets according to the power level of the same frequency in different demodulation windows. NScale [28] amplifies the time offsets between colliding packets with a novel non-stationary signal scaling strategy. CIC [27] examines the sub-symbol time domain features to cancel out interference symbols from a targeted symbol. PCube [26] resolves collisions in the phase domain by leveraging multiple antennas at a gateway. P^2LoRa [51] and NetScatter [52] aim to support concurrent transmissions of backscatter devices which send messages by backscattering LoRa signals. Similar to our work, NetScatter essentially encodes backscatter information with one-hot coding. Yet, it does not address unique practical challenges of the data aggregation in LoRa networks such as time and frequency misalignment among LoRa nodes. Unlike those works, One-shot encourages collisions to increase scalability and achieve higher aggregate data rate. To this end, One-shot designs a novel coding scheme for data aggregation, and addresses a series of practical challenges such as time and frequency misalignment among LoRa nodes.

In LoRa MAC layer, LMAC [53] supports carrier sensing for LoRa nodes with channel activity detection. S-MAC [54] models the channel access of LoRa networks as a channel scheduling problem. Unlike those works, One-shot coordinates multiple nodes to access the channel simultaneously and send coded packets concurrently.

IX. Conclusion

This paper presents a quick and accurate data aggregation method for LoRa networks. *One-shot* proposes a novel physical layer data encoding and decoding method to support a large number of LoRa nodes to send data concurrently. *One-shot* addresses technical challenges such as time and frequency misalignment among concurrent transmissions so that the aggregate data can be estimated efficiently and accurately. Experiment results show that *One-shot* can substantially outperform the state-of-the-art data aggregation method in terms of data collection efficiency as well as estimation accuracy. We believe many innovative applications can be supported by *One-shot* such as data collection over large areas, network management and maintenance, and in-field sensor data analytic.

References

[1] D. Xia, X. Zheng, F. Yu, L. Liu, and H. Ma, "Wira: Enabling cross-technology communication from wifi to lora with ieee 802.11 ax," in *IEEE INFOCOM*, 2022.

[2] F. Yu, X. Zheng, L. Liu, and H. Ma, "Loradar: An efficient lora channel occupancy acquirer based on cross-channel scanning," in *IEEE INFOCOM*. IEEE, 2022, pp. 540–549.

[3] C. Li, H. Guo, S. Tong, X. Zeng, Z. Cao, M. Zhang, Q. Yan, L. Xiao, J. Wang, and Y. Liu, "Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation," in *SenSys*, 2021, pp. 56–68.

[4] R. Li, X. Zheng, Y. Wang, L. Liu, and H. Ma, "Polarscheduler: Dynamic transmission control for floating lora networks," in *IEEE INFOCOM 2022*. IEEE, 2022, pp. 550–559.

[5] C. Li and Z. Cao, "Lora networking techniques for large-scale and long-term iot: A down-to-top survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–36, 2022.

[6] Y. Wang, X. Zheng, L. Liu, and H. Ma, "Polartracker: Attitude-aware channel access for floating low power wide area networks," *IEEE/ACM Transactions on Networking*, 2022.

[7] Y. Ren, L. Liu, C. Li, Z. Cao, and S. Chen, "Is lorawan really wide? fine-grained lora link-level measurement in an urban environment," in *ICNP*. IEEE, 2022, pp. 1–12.

[8] X. Wang, L. Kong, Z. Wu, L. Cheng, C. Xu, and G. Chen, "Slora: towards secure lora communications with fine-grained physical layer features," in *SenSys*, 2020, pp. 258–270.

[9] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on lora networking: Research problems, current solutions and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, 2019.

[10] C. Li, Z. Cao, and Y. Liu, "Deep ai enabled ubiquitous wireless sensing: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–35, 2021.

[11] X. Xia, Q. Chen, N. Hou, and y. Zheng, "Hylink: Towards high throughput lpwans with lora compatible communication," in *SenSys*, 2022, p. unknown.

[12] S. Tong, Y. He, Y. Liu, and J. Wang, "De-spreading over the air: long-range ctc for diverse receivers with lora," in *MobiCom*, 2022, pp. 42–54.

[13] X. Xia, Y. Zheng, and T. Gu, "LiteNap: Downclocking LoRa Reception," in *INFOCOM*, 2020, p. 2321–2330.

[14] X. Guo, L. Shangguan, Y. He, J. Zhang, H. Jiang, A. A. Siddiqi, and Y. Liu, "Aloba: rethinking on-off keying modulation for ambient lora backscatter," in *SenSys*, 2020, pp. 192–204.

[15] N. Hou and Y. Zheng, "Cloaklora: A covert channel over lora phy," in *ICNP*. IEEE Computer Society, 2020, pp. 1–11.

[16] 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 *SIGCOMM*, 2018, pp. 147–160.

[17] X. Guo, L. Shangguan, Y. He, N. Jing, J. Zhang, H. Jiang, and Y. Liu, "Saiyan: Design and implementation of a low-power demodulator for lora backscatter systems," in *NSDI*, 2022, pp. 437–451.

[18] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey," 2006.

[19] L. Chen, J. Xiong, X. Chen, S. I. Lee, K. Chen, D. Han, D. Fang, Z. Tang, and Z. Wang, "Widesee: Towards wide-area contactless wireless sensing," in *SenSys*, 2019, pp. 258–270.

[20] A. Gadre, F. Yi, A. Rowe, B. Iannucci, and S. Kumar, "Quick (and dirty) aggregate queries on low-power wans," in *IPSN*. IEEE, 2020, pp. 277–288.

[21] K. Yang and W. Du, "LLDPC: A Low-Density Parity-Check Coding Scheme for LoRa Networks," in *SenSys*, 2022.

[22] LoRa Alliance, "LoRaWAN for Developer," in *https://lora-alliance.org/lorawan-for-developers*, Aug. 2020.

[23] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937–947, 2010.

[24] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *SIGCOMM*, 2017, pp. 309–321.

[25] X. Xia, Y. Zheng, and T. Gu, "Ftrack: Parallel decoding for lora transmissions," *IEEE/ACM Transactions on Networking*, vol. 28, no. 6, pp. 2573–2586, 2020.

[26] X. Xia, N. Hou, Y. Zheng, and T. Gu, "Pcube: scaling lora concurrent transmissions with reception diversities," in *MobiCom*, 2021, pp. 670–683.

[27] M. O. Shahid, M. Philipose, K. Chintalapudi, S. Banerjee, and B. Krishnaswamy, "Concurrent interference cancellation: decoding multi-packet collisions in lora," in *SIGCOMM*, 2021, pp. 503–515.

[28] S. Tong, J. Wang, and Y. Liu, "Combating packet collisions using non-stationary signal scaling in lpwans," in *MobiSys*, 2020, pp. 234–246.

[29] Z. Xu, P. Xie, and J. Wang, "Pyramid: Real-time lora collision decoding with peak tracking," in *IEEE INFOCOM*. IEEE, 2021, pp. 1–9.

[30] C. Li, X. Guo, L. Shangguan, Z. Cao, and K. Jamieson, "Curvinglora to boost lora network throughput via concurrent transmission," in *NSDI*, 2022, pp. 879–895.

[31] N. Hou, X. Xia, Y. Wang, and Y. Zheng, "One shot for all: Quick and accutate data aggregation for lpwans," in *IEEE INFOCOM*. IEEE, 2023.

[32] N. A. A. Ali and N. A. A. Latiff, "Environmental monitoring system based on lora technology in island," in *ICSigSys*. IEEE, 2019, pp. 160–166.

[33] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *IEEE Cat. No.05EX960*. IEEE, 2005, pp. 108–120.

[34] S. Qi, Y. Lu, Y. Zheng, Y. Li, and X. Chen, "Cpds: Enabling compressed and private data sharing for industrial internet of things over blockchain," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2376–2387, 2020.

[35] Q. Chen and J. Wang, "Aligntrack: Push the limit of lora collision decoding," in *ICNP*. IEEE, 2021, pp. 1–11.

[36] N. Hou, X. Xia, and Y. Zheng, "Don't miss weak packets: Boosting lora reception with antenna diversities," in *IEEE INFOCOM*. IEEE, 2022, pp. 530–539.

[37] J. Liu, J. Gao, S. Jha, and W. Hu, "Seirios: leveraging multiple channels for lorawan indoor and outdoor localization," in *MobiCom*, 2021, pp. 656–669.

[38] D. Guo, C. Gu, L. Jiang, W. Luo, and R. Tan, "Illoc: In-hall localization with standard lorawan uplink frames," *IMWUT*, vol. 6, no. 1, pp. 1–26, 2022.

[39] N. Hou, X. Xia, and Y. Zheng, "Jamming of lora phy and countermeasure," in *IEEE INFOCOM*, 2021, pp. 1–10.

[40] Z. Xu, P. Xie, J. Wang, and Y. Liu, "Ostinato: Combating lora weak links in real deployments," in *ICNP*. IEEE, 2022, pp. 1–11.

[41] S. Tong, Z. Shen, Y. Liu, and J. Wang, "Combating link dynamics for reliable lora connection in urban settings," in *MobiCom*, 2021, pp. 642–655.

[42] T. K. Goyal and V. Sahula, "Lightweight security algorithm for low power iot devices," in *ICACCI*. IEEE, 2016, pp. 1725–1729.

[43] S. Randhawa and S. Jain, "Data aggregation in wireless sensor networks: Previous research, current status and future directions," *Wireless Personal Communications*, vol. 97, no. 3, pp. 3355–3425, 2017.

[44] I. Atoui, A. Ahmad, M. Medlej, A. Makhoul, S. Tawbe, and A. Hijazi, "Tree-based data aggregation approach in wireless sensor network using fitting functions," in *ICDIPC*. IEEE, 2016, pp. 146–150.

[45] F. Xiangning and S. Yulin, "Improvement on leach protocol of wireless sensor network," in *SENSORCOMM*. IEEE, 2007, pp. 260–264.

[46] P. William, A. Badholia, V. Verma, A. Sharma, and A. Verma, "Analysis of data aggregation and clustering protocol in wireless sensor networks using machine learning," in *ICECMSN*. Springer, 2022, pp. 925–939.

[47] M. Zimmerling, L. Mottola, and S. Santini, "Synchronous transmissions in low-power wireless: A survey of communication protocols and network services," *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–39, 2020.

[48] Z. Wang, L. Kong, K. Xu, L. He, K. Wu, and G. Chen, "Online concurrent transmissions at lora gateway," in *IEEE INFOCOM*. IEEE, 2020, pp. 2331–2340.

[49] X. Wang, L. Kong, L. He, and G. Chen, "mlora: A multi-packet reception protocol in lora networks," in *ICNP*. IEEE, 2019, pp. 1–11.

[50] S. Tong, Z. Xu, and J. Wang, "Colora: Enabling multi-packet reception in lora," in *IEEE INFOCOM*, 2020, pp. 2303–2311.

[51] J. Jiang, Z. Xu, F. Dang, and J. Wang, "Long-range ambient lora backscatter with parallel decoding," in *MobiCom*, 2021, pp. 684–696.

[52] M. Hessar, A. Najafi, and S. Gollakota, "Netscatter: Enabling large-scale backscatter networks," in *NSDI*, ser. NSDI'19. USA: USENIX Association, 2019, p. 271–283.

[53] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "Lmac: Efficient carrier-sense multiple access for lora," in *MobiCom*, 2020, pp. 1–13.

[54] Z. Xu, J. Luo, Z. Yin, T. He, and F. Dong, "S-mac: Achieving high scalability via adaptive scheduling in lpwan," in *INFOCOM*, 2020, p. 506–515.