

Processor-Sharing Internet of Things Architecture for Large-scale Deployment

Qianhe Meng

University of Electronic Science and
Technology of China
qianhe@std.uestc.edu.cn

Han Wang

University of Electronic Science and
Technology of China
wang_han@std.uestc.edu.cn

Chong Zhang

University of Electronic Science and
Technology of China
Southwest Petroleum University
zhangchong92@swpu.edu.cn

Yihang Song

University of Electronic Science and
Technology of China
songyihang@uestc.edu.cn

Songfan Li

The Hong Kong University of Science
and Technology
lisongfan@ust.hk

Li Lu*

University of Electronic Science and
Technology of China
luli2009@uestc.edu.cn

Hongzi Zhu

Shanghai Jiao Tong University
hongzi@sjtu.edu.cn

ABSTRACT

Large-scale IoT sensor deployment calls for inexpensive, low-power sensor nodes that still perform long-range, large-scale networking at the system level. However, current sensor nodes are constructed according to the 'one-size-fits-all' embedded design, where the processor and RF transceiver are indispensable but underutilized in low-duty cycles, resulting in overwhelmingly significant unit price and run-time power. In this paper, we propose a novel processor-sharing IoT architecture that converts the vast majority of sensor nodes from embedded computers to low-end RF peripherals. The conventional full-fledged sensor nodes are smashed into the air, and the scattered chips are scaled well with negligible overheads through a virtual I²C bus called *RFBus*. Specifically, the *RFBus* interface is designed to be backward compatible with the I²C bus interface, and thus, the *RFBus* network inherits versatile link layer services transparently from the well-established I²C link layer protocol. We design the *RFBus* with a joint consideration of system-level performance and deployment costs and evaluate the prototypes in indoor and outdoor scenarios. The result indicates that the proposed architecture achieves $6.09 \times$ (indoor) and $6.69 \times$ (outdoor) energy saving and reduces the unit price of sensor nodes by 23.5% (indoor) and 33.5% (outdoor).

*Li Lu is the corresponding author to this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SENSYS '24, November 4–7, 2024, Hangzhou, China

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0697-4/24/11

<https://doi.org/10.1145/3666025.3699333>

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; • **Hardware** → **Communication hardware, interfaces and storage**.

KEYWORDS

Processor-sharing architecture, Virtual I²C bus, Large-scale sensor deployment

ACM Reference Format:

Qianhe Meng, Han Wang, Chong Zhang, Yihang Song, Songfan Li, Li Lu, and Hongzi Zhu. 2024. Processor-Sharing Internet of Things Architecture for Large-scale Deployment. In *The 22nd ACM Conference on Embedded Networked Sensor Systems (SENSYS '24)*, November 4–7, 2024, Hangzhou, China. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3666025.3699333>

1 INTRODUCTION

A wide range of Internet of Things (IoT) applications, such as wild-fire detection and structural health monitoring, are keen to deploy sensor nodes at a large scale to predict or identify natural disasters, reducing or avoiding significant damage and death. For instance, Maui wildfires [1] have caused 101 people to be killed and almost 3,000 structures destroyed in August 2023. However, it is still challenging today to realize the large-scale deployment of IoT sensor nodes because of the following two factors.

• *Node's manufacturing cost is unaffordable.* Current commercial off-the-shelf (COTS) sensor node is deployed in conjunction with a processor and radio frequency (RF) transceiver. As a result, the unit price of such an embedded computer is generally in the range of a few dollars to tens of dollars, which is overwhelmingly expensive for large-scale deployments. The manufacturing cost of current full-fledged sensor nodes loses fit with the value of the 'Things' they sense, e.g., temperature.

• *Node's maintenance cost is unaffordable.* RF transceiver is the node's primary energy consumer, which drains dozens of milliwatts (mW). A button battery, for example, can only support such

a transceiver for a few hours. Although the node's lifetime may be extended to years by practicing an extremely low duty cycle, the latency becomes excessive in trade-off and unacceptable for IoT sensing. It is nontrivial to change batteries frequently for a vast number of sensor nodes, not to mention that some of them may be deployed in hard-to-reach locations, *e.g.*, treetops.

A consensus for designing sensor nodes with the potential of large-scale deployment is to offload some of the nodes' functionality to a powerful gateway, mitigating the above mentioned cost issues. Significant effort has been made by utilizing backscatter communication techniques [2–4], replacing superheterodyne receiver by envelope detector and piggybacking sensor data via RF carriers that generated by the gateway. Though node's power-hungry transceiver is replaced with μW front-end, the comprehensive communication range is typically limited to 30 meters, requiring more gateways for large-scale deployments. Their deployment costs are increased rather than decreased at the system level.

Another direction considers shifting the node's computation burden to the gateway [5–8], relaying on the gateway to replicate the processor's functionality through RF communication. These sensor nodes are relieved from sensor-related computation workload, *e.g.*, chip access control; however, RF-related computation workload, *e.g.*, link layer control, still needs to be executed on node. Otherwise, only a point-to-point network between the gateway and a sensor node can be formed. Such demand requires additional discrete logic components or even field programmable gate arrays (FPGA) to be embedded on the 'simplified' node. It is hard to say that the unit price of such sensor nodes is reduced. Moreover, the RF communication overhead is significantly increased. Thus, they can only work with ultra-low-power but short-range RF communication methods with terrible network scalability. Otherwise, the cost of computation offload will outweigh the benefit. As a result, no existing successful scheme, to the best of our knowledge, can tackle this long-standing issue.

For the ease of large-scale deployment, we argue that the functionality of the sensor nodes should not be offloaded to gateways. Ostensibly, the sensor nodes are simplified by such a communication/computation offload. However, the network performance, *e.g.*, coverage and scalability, is sacrificed by the compromise made by gateways, turning this offload scheme into a trade-off dilemma and thus increasing the deployment costs at the system level.

In contrast, we appeal for more efficient utilization of the on-node processor and RF transceiver, and wonder if a full-fledged sensor node can be smashed into the air for broader sensor coverage. Imagine if a full-fledged node could have access to its neighboring sensor chips with negligible overheads. These chips can consolidate a spatially distributed yet centrally processed sensor cluster. The sensor coverage of the embedded computer will be extended from a fixed point to a certain area, inferring that fewer full-fledged nodes, *i.e.*, cluster heads, need to be deployed in field. As a bonus, the gateway will only schedule a small number of cluster heads, inspiring its potential for larger-scale sensor networking.

In this paper, we propose a novel processor-sharing architecture with a hybrid IoT network. As illustrated in Fig. 1, the vast majority of sensor nodes, *i.e.*, lite nodes, are simplified without processor or power-hungry RF transceiver, playing as low-end *RF peripherals* rather than standalone embedded computers; meanwhile, the rest

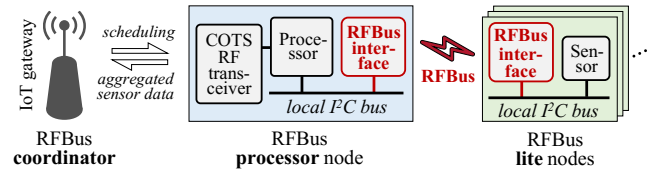


Figure 1: A processor-sharing IoT architecture that consolidates two heterogeneous types of sensor nodes via RFBus.

of the sensor nodes, *i.e.*, processor nodes, are fully fledged. To interconnect these two heterogeneous sensor nodes with negligible overheads, we design a *virtual* Inter-Integrated Circuit (I^2C) bus communication scheme, referred to as *RFBus*. *RFBus* combines the local I^2C buses on the processor and lite nodes into a unified I^2C network by piggybacking I^2C bus signals via RF carriers. As a result, the processor can directly access inter-node sensors via the *RFBus* network as if these are local sensors.

The sweet spot of the *RFBus* network design is that it leverages the well-established link layer protocol integrated with the I^2C bus interface. Specifically, the *RFBus* network transparently inherits versatile I^2C 's link layer services, *e.g.*, addressing, anti-collision and reliable delivery, by the designed *RFBus* interface that is backward compatible with the I^2C bus interface. The need for an on-node link layer-protocol executor and other 'glue logic' is eliminated from the lite nodes, making these nodes extremely cheap and power-friendly in potential.

Moreover, the proposed *RFBus* networks that consolidate the scattered processors and their neighboring sensor chips are considered integrated by an already existing network and scheduled by a remote gateway. The choice of such an upper-level network depends on the sensing application and the RF compatibility of the gateway, providing design flexibility to the system users. LoRaWAN might be an excellent option for outdoor applications, *e.g.*, wildfires detection; meanwhile, WiFi or WiFi Halow with enhanced communication range (around 800 meters) is more suitable for indoor applications, *e.g.*, structural health monitoring.

In a brief review of the processor-sharing IoT architecture, sensor data is firstly obtained by the distributed sensor clusters through multiple *RFBus* networks, then aggregated at clusters' centralized processors, and finally gathered to a remote gateway by utilizing the COTS RF transceivers on the cluster heads, *i.e.*, processor nodes.

This paper addresses two key challenges in implementing this design. Ultimately, we hand out an inexpensive, low-power *RFBus* interface that is also backward compatible with the I^2C bus interface, and establish a chip-oriented multiple-to-multiple *RFBus* network for transparent, large-scale inter-node chip access. It is worthwhile to note that the lite nodes are carefully designed to be minimalist by some counterintuitive but ingenious ruminations.

Challenge 1: Piggybacking ambiguous I^2C signals via RF carriers with negligible overheads. I^2C bus interface is specified with the *open-drain* output, where ternary bus information is ultimately expressed in binary signal voltage levels. Specifically, the high voltage-level I^2C data, *i.e.*, SDA, signal expresses not only logic '1' but also a unique I^2C bus state of *idle*. Given that I^2C bus remains idle when released, the *RFBus* interface must determine if it should

generate RF carriers when the SDA remains ambiguous at a high voltage level. Otherwise, the virtual I²C bus communication fails in signal collisions. To address this by minimalist design principle, in § 3.1, we design an RF open-drain output to express the open-drain outputs in the RF medium without parsing the idle bus state. On the other hand, I²C interfaces avoid bus collision by listening to the bus wire continuously, a similar Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) procedure. In § 3.2, we incorporate an ultra-low-power RF demodulator to make the power-constrained sensor nodes continuously receive RF carriers, supporting such a CSMA/CA procedure in the physical layer. At the same time, an RF modulator with transparent transmission capability is introduced to broadcast and synchronize multiple local I²C buses in the RF medium under the timing constraint of the I²C bus protocol, ramming the physical layer foundation of the virtual I²C bus network.

Challenge 2: Agnostic rhythm of half-duplex I²C bus communication and cross-talk among lite nodes. In standard I²C bus session, I²C master and slave chips transfer data in a half-duplex manner, where their bus interfaces hold I²C bus lines alternately for bi-directional communication. However, such communication rhythm is transparently conducted by I²C's intellectual property (IP) core that is integrated with each I²C bus interface. The flow direction of bus data is agnostic to the lite nodes without a processor or any other protocol parser. Such agnostic rhythm will certainly lead to a deadlock problem in the RFBus network, where local I²C buses on all processor and lite nodes are clamped at low voltage levels. To resolve it, in § 4.1, we shift the lite node's RF signaling control to the processor node with no extra RF communication overhead or hardware introduced to the lite nodes. On the other hand, multiple lite nodes are logically parallel-connected in the virtual I²C bus network, where cross-talk issues will incur if further RF media access control among them is absent. For simplicity, in § 4.2, we design an RFBus frame that comprises a 2-byte frame head ahead the standard I²C frame, and implement a decoder on the lite node with only five logic components introduced.

We implement and evaluate RFBus prototypes, conducting comprehensive evaluations in indoor and outdoor scenarios. In configurations where nine lite nodes concurrently share a processor node, the results indicate that RFBus facilitates substantial energy savings — achieving a $6.09 \times$ increase indoors and a $6.69 \times$ increase outdoors. Moreover, this approach significantly reduces the cost per node by 23.5% compared to WiFi nodes and by 33.5% relative to LoRa nodes. These results underscore the potential deployment benefits of adopting the processor-sharing architecture within the IoT ecosystem. These benefits are anticipated to magnify with the expansion of deployment scales.

Contributions are summarized as follows.

Computation-communication co-optimization. We optimize the computation and communication overheads of sensor deployments at the system level. The COTS processor and RF transceiver are used more efficiently by sharing among neighboring lite nodes and uploading aggregated sensor data with less communication overhead.

Processor-sharing IoT architecture. An architecture that turns the majority of sensor nodes into low-end RF peripherals rather than standalone embedded computers, reducing the overall

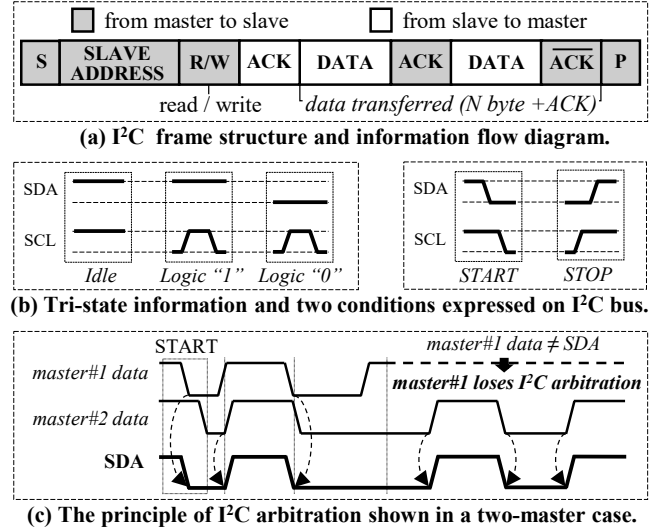


Figure 2: I²C's (a) frame structure, (b) signal expression, and (c) multi-master arbitration.

manufacturing and maintenance costs for large-scale deployment with potential enhancement on network performance.

I²C compatible RF interface. The method of piggybacking I²C bus signals via RF carriers bridges the gap between wired bus and wireless communication. More importantly, it inherits versatile link layer services from the I²C bus protocol, eliminating the need to execute link layer protocol on most sensor nodes. With this, these nodes can be extremely minimalist.

2 I²C PRIMER

About 73% of the digital sensor chips sold at the international electronic component distributor Digi-Key[®] adopt the I²C bus interface. I²C's prosperity lies in its minimalist physical layer design and rich functionality defined by its well-established link layer protocol.

Link layer protocol. I²C bus is designed with a multiple-to-multiple link layer protocol, which is executed by the I²C's IP core that integrated with each I²C bus interface. As shown in Fig. 2(a), I²C bus data is packetized in a tidy frame, and chips hold the bi-directional bus lines alternately in each bus session. Specifically, the master chip, *i.e.*, processor, initiates the bus session via START condition, exclusively occupies I²C after the addressing is acknowledged, and terminates the bus session with STOP condition after sensor data is received. Cyclic Redundancy Check (CRC) bits are integrated into the payload for some sensor chips, making the I²C communication robust to the interference.

Open-drain output and 'wired-AND.' I²C bus interface is specified to be driven by the open-drain output and performs a unique electrical property of 'wired-AND.' Specifically, suppose two or more I²C chips are connected in the same bus. In that case, any chip's interface that holds the bus line will short-circuit the data wire to the logic ground, and other interfaces cannot transfer data until the bus is released. It is likewise the action of an 'invisible' AND gate, which is the physical layer foundation of multiple I²C

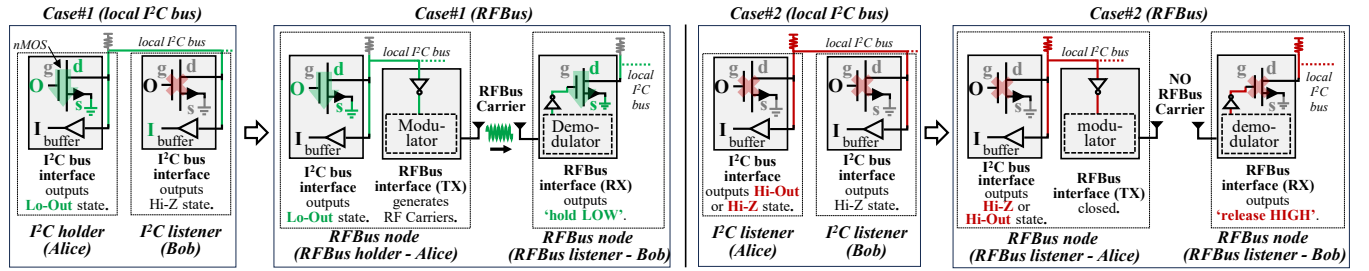


Figure 3: The diagram shows how the RFBUS interface piggybacks I^2C 's tri-state bus information via RF carriers while preserving the electrical property of the 'wired-AND'. For convenience, we present two RFBUS nodes, called *Alice* and *Bob*. In Case#1, *Alice* is the bus holder whose output expresses the Lo-Out state. Her RFBUS interface (TX) generates an RF carrier that is received by *Bob*'s RFBUS interface (RX). The drain (shortened as 'd') of *Bob*'s RFBUS interface (RX) is connected to the logic ground and holds his local I^2C bus wire at a low voltage level. At a high level, the RFBUS interface (RX) behaves as an agent of *Alice* at *Bob*. In Case#2, *Alice* and *Bob* are bus listeners who release the bus. In this case, the RFBUS carrier is not generated, so the drain of the RFBUS interface (RX) is floated and pulled up naturally by the resistor.

properties, e.g., multiple-to-multiple access, and collision arbitration.

Anti-collision, arbitration, and tolerance. I^2C is an actual multi-master bus, including collision avoidance and arbitration, to prevent data corruption if two or more masters simultaneously initiate data transfer. Specifically, all I^2C interfaces listen to the bus wire continuously, and master devices initiate new bus sessions only when the bus is free. It is likewise a CSMA/CA procedure. However, two or more masters may generate valid START commands at the same time; see Fig 2(c). In this circumstance, I^2C protocol arbitrates between multiple bus sessions, and the ones that lose the arbitration will cancel their transmission. It is worth noting that multiple masters may complete their sessions without error if their data is the same. Such a collision is tolerated.

Organization of the remainder of this paper. In the following sections, we focus on two technical obstacles of RFBUS. First, in the physical layer, we describe the technical design of the RFBUS interface that is fully backward compatible with I^2C bus interface in § 3. Second, in the link layer, we present how RFBUS becomes a multi-to-multi RF network; the technical design is described in § 4. In the remainder of this paper, we present the implementation, evaluation, related work, and discussion, and finally give our conclusions.

3 RFBUS INTERFACE

In this section, we introduce the RFBUS interface that can piggyback I^2C signals via RF with negligible overheads. We first design an RF open-drain output that expresses I^2C bus information in the RF domain. Then, an RF modulation-demodulation scheme is discussed to interconnect nodes with minimized deployment costs at the system level.

3.1 RF Open-drain Output

I^2C bus expresses ternary information through binary voltage levels. Despite data information of logic '1' and logic '0', I^2C has a unique bus state called *idle*, inferring that all chips release the bus with no ongoing transmission. The ensuing problem is that the SDA remains high in logic '1' and the idle state. The ambiguous high SDA should

be distinguished as I^2C signals are designed to be piggybacked among the processor and lite nodes.

The I^2C bus interface distinguishes these two states by utilizing its integrated IP core, parsing the context of the I^2C signals. See the left part of Fig. 2 (b), a pulse is provided on the I^2C clock, i.e., SCL, to read high SDA as logic '1'; while the voltage levels of both SDA and SCL remain high when the bus is released. Technically, we may distinguish logic '1' from the idle state by detecting the SCL pulse when SDA remains high. However, implementing such a design requires an additional FSM, i.e., logic components, to be embedded on the lite nodes, violating our minimalist design principle.

We wonder if it is possible to piggyback I^2C signals without distinguishing the ambiguous high SDA. The answer is positive as we observe that the ternary expression of the I^2C bus information is realized by the binary physical layer control. Specifically, the tri-state open-drain output integrated with I^2C interface is realized by conducting or cutting off the n-channel metal-oxide-semiconductor (nMOS). As depicted in Case#1 (wired I^2C) of Fig. 3, when an I^2C chip holds the SDA and expresses logic '0', the n-channel of its nMOS is conducted, and thus the drain short circuits the SDA to the logic ground. *It is likewise an invisible AND gate, called 'wired-AND.'* On the other hand, as depicted in Case#2 (wired I^2C), when the bus is released by both interfaces that express idle or hold by one interface to express logic '1', the gates of their nMOSs are set at low, and the cutoff n-channels make the drains float. *This is where the name of 'open-drain' comes.* In this case, the I^2C bus wire will be set at a high voltage level with the help of a pull-up resistor that is serially connected between the drains and the power supply. Overall, I^2C bus interface generates signals through the binary nMOS control, i.e., 'release HIGH' and 'hold LOW'. There is no distinction between logic '1' and idle at the physical layer.

Inspired, we design an RF open-drain output that only piggybacks low voltage level I^2C signals without distinguishing the ambiguous high voltage level signals. This design is implemented by a 1-bit inverter for signal conversion, an RF modulator (TX) for RF carrier generation, and an RF demodulator (RX) designed with open-drain output for reception. Fig. 3 Case#1 (RFBUS) illustrates the operation principle of the RF open-drain in a case where the

processor node transmits logic '0' to the lite node. The processor node's TX generates RF carriers as its local SDA remains low, and then the RX of the lite node recovers the received "logic '0'" by short-circuiting its local SDA to the logic ground. In another case, Fig. 3 Case#2 (RFBUS), where the processor node's local SDA is pulled up, its on-node TX will remain silent. Thus, the inter-node SDAs are pulled up by the combination of high-impedance RX output and the pull-up resistor. I²C's 'wired-AND' functionality is preserved in the RF medium with the form of 'wireless-NOR', ramming the physical layer foundation for the virtual I²C network to inherit versatile I²C's link layer services transparently.

3.2 RF Front-End Design

We design a hybrid RF front-end to construct a symmetric bi-directional RF communication link between the processor and lite nodes. The design considerations and details are as follows.

Principle 1: Near-zero power RF reception on the lite nodes. I²C bus operates in the request-respond mode, where the bus slave, *i.e.*, lite node, must listen to the bus continuously, and then respond to the bus master, *i.e.*, processor node, in time constraints. The lite node must have RF carrier sensing capability, otherwise the bus session will be missed or responded to with a timeout. Therefore, the RF reception power consumption of the lite node needs to be minimized to near-zero level to prolong its lifetime.

Principle 2: Balanced deployment costs and communication performance. The virtual I²C bus network is built among the processor and lite nodes that are assumed to be largely deployed in field. Their RF front-ends should be designed with balanced deployment costs, *i.e.*, manufacturing and maintenance costs, and the inter-node communication performance. Otherwise, the system-level deployment costs will not be kept at a low level, making the network impractical and inefficient in the concept of processor sharing among neighboring IoT sensor nodes.

3.2.1 Lite node's RF front-end. Demodulation. Conventional superheterodyne receivers down-convert the RF signal to an intermediate frequency (IF) signal and then coherently demodulate the sampled IF signal by mixing it with a synchronized oscillator (LO) signal. A series of power-hungry components are necessary, like a crystal oscillator (XO) and a phase-locked loop (PLL). At a high level, the power consumption is traded for higher sensitivity, making these RF receivers unaffordable for power-constrained nodes, especially the ones with RF carrier sensing demands. Instead, passive receivers enjoy the μ W-level power benefit by adopting a non-coherent demodulation technique called envelope detection. By detecting the energy level of received RF signals, the envelope detector can down-convert and demodulate the RF signal passively without any high-power components, *e.g.*, XO and PLL. Meanwhile, the manufacturing cost of the passive receivers is cheaper than that of the superheterodyne receivers. Although the sensitivity of such receivers is poor, their ultra-low power characteristics make it possible to perform carrier listening on power-constrained nodes. We fuse the RF open-drain design with the passive demodulation technique and provide an RF open-drain receiver with ultra-low power consumption. As shown in Fig. 4, the received RF signal is first bandpass filtered, then down-converted and demodulated passively by envelope detection. The obtained analog baseband

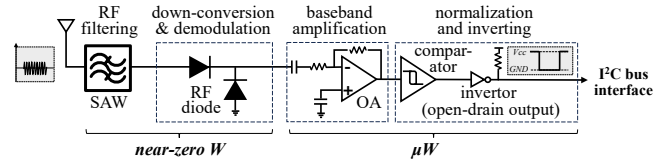


Figure 4: The block diagram and signal processing flow of the RF open-drain receiver with ultra-low power consumption.

is amplified and normalized by an operational amplifier (OA) and comparator with open-drain output, respectively.

Modulation. In the field of low-power RF design, backscatter communication is typically used in conjunction with the envelope detector to build an ultra-low-power RF front-end on the power-constrained devices. However, this paradigm only holds if one side of the communication is a power-constrained device, *e.g.*, RFID tag, and the other is a powerful device, *e.g.*, RFID reader. At a high level, the communication burden is not eliminated from the system but shifted from one device to another. Contrary to this assumption, both sides of the RFBUS communication are power-constrained nodes. If the lite node backscatters its local I²C bus signal to the processor node, the processor node will be bulky and power-consuming. The deployment costs of the processor and lite nodes will lose fit, and the system-level deployment costs will increase for the following reasons. First, the processor node will need to receive the backscattered signal while emitting the backscatter carrier; expensive RF isolation circuits will be necessarily needed for such a full-duplex functionality [9, 10]. Second, the backscattered signal is feeble after two-way propagation; thus, a sensitive but power-hungry receiver must be deployed on the processor node to counter interference. Moreover, the backscatter communication range is typically limited to 30 meters; it is non-trivial to manufacture and maintain the assumed expensive power-hungry processor nodes in scale. Upon these, we consider adopting an active transmitter to the lite node. Ostensibly, the RF power consumption is not fully optimized on the lite node as active transmitter is more expensive and power-consuming than the backscatter; however, the deployment costs and communication range of RFBUS communication is optimized at the system level.

3.2.2 Processor node's RF front-end. Modulation. To cope with the envelope detector used by the lite nodes, the most straightforward and cost-efficient approach is to generate amplitude-modulated (AM) signals on the processor node. Though AM signals suffer from poor interference immunity, this physical layer weakness can be mitigated through link layer services provided by I²C protocols, *e.g.*, ACK and CRC. Note that frequency shifting keying (FSK) or phase shift keying (PSK) based modulation seems to be a better option as frequency and phase are more robust to the interference. However, the envelope detector cannot obtain the processing gain of either FSK or PSK modulation as it demodulates the RF signal according to its amplitude rather than frequency or phase. To the best of our knowledge, the processing gain of FSK and PSK modulation can only be obtained in two ways. One direction is to use the power-hungry superheterodyne receivers to actively down-convert and demodulate the FSK/PSK signals on the lite node. Another direction

is to emit a synchronized RF helper signal along with the FSK modulated signal [11–13], which largely increases the complexity and power consumption of the processor node. With these in mind, an on-off-keying (OOK) modulator is used on the processor node to piggyback the envelope information of digital bus signals by AM signals. This design satisfies our target of consolidating neighboring sensor nodes rather than remote ones and balances the deployment costs and communication performance at the system level.

Demodulation. The processor node only needs to activate its receiver and perform the I²C-defined CSMA/CA procedure before it initiates a bus session, without the need to listen to the carrier as the lite node does continuously. Therefore, it seems reasonable to deploy a superheterodyne receiver for high reception sensitivity. However, the sensitivity of the lite node's non-coherent demodulation is poor, and employing coherent demodulation on the processor node will result in asymmetric uplink and downlink communication ranges between the processor and lite nodes. It is energy inefficient to prolong the uplink communication range from the lite node to the processor node while leaving the downlink communication range to remain the bottleneck of the bi-directional communication between the processor and lite nodes. For this reason, the envelope detector is also used on the processor node to make the bi-directional, inter-node communication symmetric and energy efficient at the system level.

4 RFBUS SIGNALING AND NETWORKING

This section starts with a case where only one processor node and one lite node exist in the RFBus network, then RFBus is extended to a multiple-to-multiple network.

4.1 Half-Duplex RF Signaling

In each I²C session, a bus master and an addressed bus slave hold I²C bus alternately for bi-directional data transfer. The communication rhythm is defined by the I²C protocol and executed by the I²C IP core integrated with each I²C interface. By parsing the context of I²C signals, the I²C bus interface can determine the timing of transferring data onto the I²C bus. However, the lite node is not embedded with a processor or FSM that can parse the context of I²C signals - the execution of I²C protocol is transparent to the lite node. Consequently, the lite node cannot control its RFBus interface to generate RF carriers only when its on-node sensor chip holds the local I²C bus.

In that case, its RFBus interface can only mechanically generate RF carriers to piggyback ALL I²C signals on its local I²C bus. It is wrong because some of these local I²C signals are initially generated by the processor node and then received by the lite node. They should not be carried back to the processor node. Otherwise, the virtual I²C bus will fail in the deadlock problem. Specifically, as we introduced in the last section, the I²C device holds the bus by connecting its open-drain output to the logic ground. Imagine a processor accessing an inter-node sensor chip at a particular moment; RF carrier will be generated by its RFBus interface (TX) and received by the lite node's RFBus interface (RX). Then, the lite node's local I²C bus will be grounded by its RFBus interface (RX) and express logic '0'. However, the lite node's RFBus interface (TX) cannot identify the logic '0' generated by the inter-node processor;

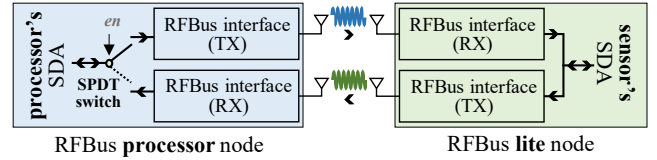


Figure 5: The RFBus interface of a lite node is directly connected to the node's local I²C bus, while that of the processor node is connected to an SPDT switch, which is controlled according to the communication rhythm of I²C.

it mechanically generates an RF carrier to piggyback the logic '0'. Consequently, the local I²C bus of the processor node will be grounded, and the above procedure will be repeated infinitely. At a high level, the local I²C bus of both processor and lite nodes will be clamped at the low voltage level, leading to a deadlock problem in the RFBus network.

Our basic idea is to interrupt the continuous 'grounding' operation in the virtual I²C bus. First, the processor and lite nodes' RFBus interface (TX) are set to generate RF carriers with different frequencies, establishing a bi-directional RF traffic. Then, we leave the lite node continuously piggybacking its local I²C signal over RF carriers. However, the processor node only receives the RF carriers whenever the addressed sensor chip embedded at the lite node is expected to hold the bus line. As illustrated in Fig. 5, we implement a simple hardware solution by placing a single-pole double-throw (SPDT) switch between the processor and its RFBus interface. The SPDT switch is controlled based on the rhythm of I²C communication that is output by the processor. When I²C session is executed to the point when the processor should hold the I²C bus, the SPDT switch connects the pin of the processor node's local I²C bus to the input of the RFBus interface (TX). Otherwise, the switch connects the pin to the processor node's RFBus interface (RX) output to receive the RF carrier.

4.2 Multiple-to-Multiple RFBus Network

So far, a point-to-point communication link has been constructed between a processor node and a lite node. In this subsection, we extend it to a multiple-to-multiple network, where multiple processor and lite nodes co-exist in the same virtual I²C bus. Recall § 2, collisions between the I²C bus masters, *i.e.*, the processor nodes, are eliminated by the arbitration procedure of I²C protocol. In the following part, we focus on the collisions between lite nodes.

Lite nodes should only respond to the processor node whenever their on-node sensor chip is addressed in the ongoing I²C bus session. Otherwise, lite nodes will transmit signals in cross-talk, resulting in noisy RF channels with traffic jams. To tackle this, a straightforward approach is to activate the lite node's RFBus interface whenever their on-node sensor chip is addressed by I²C chip address. However, the I²C address decoder is integrated with the bus interface. An FSM must be additionally implemented on each lite node to parse I²C frame. The FSM should parse the I²C START condition and its following I²C address to initiate the RFBus interface and terminate when the corresponding STOP condition is detected. Such FSM could be expensive and power-consuming

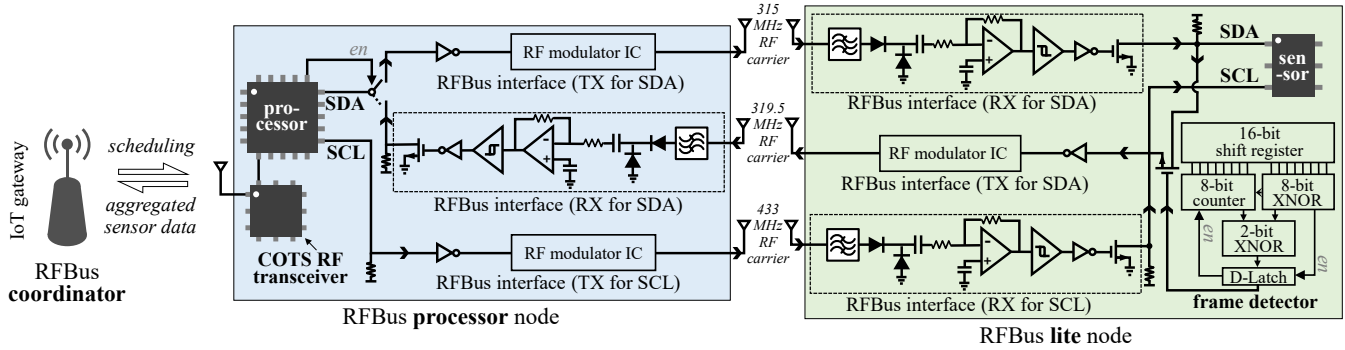


Figure 6: The schematic diagram of the processor and lite nodes is illustrated with the data flow of the RFBUS network.

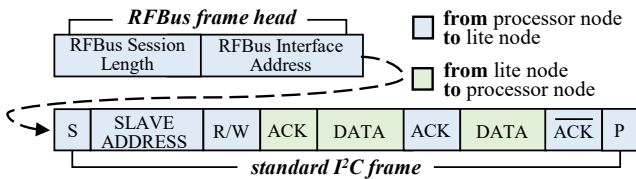


Figure 7: The RFBUS frame comprises a 2-byte frame head ahead of the standard I²C frame.

because the START and STOP conditions are constructed by a joint sequential combination of SDA and SCL; see the right part of Fig. 2 (b). Moreover, such timing sequences vary along with the bus speed, e.g., the minimum hold time of START condition is 4 μ s in standard mode but 0.6 μ s in fast mode. More importantly, such design defaults our design purpose of simplifying lite nodes into low-end RF peripherals rather than standalone devices.

Alternatively, we design a 2-byte RFBUS frame head and program the processor node to transfer it before initiating a new I²C bus session. As depicted in Fig. 7, 1-byte 'RFBUS session length' defines the length of the incoming I²C frame. Another byte of 'RFBUS interface address' activates a specific RFBUS interface connected with the addressed sensor chip of the incoming I²C session. Since the I²C bus interface only starts processing the input signals whenever a START condition is detected, this preceding 2-byte RFBUS frame head is naturally ignored without interrupting the I²C sessions.

As illustrated in Fig. 8, we design a decoder that consists of only five logic elements to parse the RFBUS frame head and further activate and shut down the RFBUS interface (TX) on each lite node. The received RFBUS frame head first passes through an address detection circuit that consists of a shift register and a latch. If the received 'RFBUS interface address' is matched with the one predefined by the 8-bit Dual In-line Package (DIP) switch on the lite node, the RFBUS interface (TX) will be activated and connected to the local I²C bus. At the same time, a counter starts to count down the number defined by 1-byte 'Length.' Finally, at the moment when the counter is cleared to zero, the current I²C session is finished, so the RFBUS interface (TX) is disconnected from the local I²C bus and shut down.

As a bonus, the 'RFBUS interface address' can enlarge the addressing space of the RFBUS network. It is useful when multiple sensor

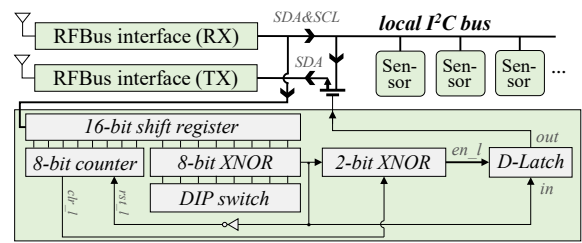


Figure 8: The block diagram of the RFBUS frame decoder on the lite node. The RFBUS interface (TX) is connected with the lite node's local I²C bus only when the inter-node processor addresses its on-node sensor chip.

chips produced by the same company are deployed in the field. Because the most significant 4 bits of the 7-bit I²C chip address is defined as manufacturer ID, only the remaining 3 bits can be hardware defined by users. By this design, the number of accessible sensor chips within the RFBUS network is 2⁸ times 2³ equals 2,048, which is a number that can even fully satisfy the dense deployment scenario.

5 PUT THINGS TOGETHER

We put all the design details in a schematic diagram to give an overview of the proposed processor-sharing IoT architecture. RFBUS network comprises two heterogeneous types of nodes, i.e., processor node and lite node. For convenience, we present one processor node and one lite node in the diagram; see Fig. 6. The processor node is the head of the RFBUS sensor cluster, which is scheduled by a remote gateway through COTS RF communications, e.g., LoRa.

The processor node is fully-fledged with an embedded processor, a COTS RF transceiver, I²C sensor chips, and the RFBUS interface introduced in the last section. In contrast, the lite node is minimalist without a processor or COTS RF transceiver. Its on-node sensor chips are parallel-connected with the RFBUS interface through local I²C bus.

First, the processor node communicates with the IoT gateway through its on-node COTS RF transceiver, e.g., LoRa SX1276. Sensing tasks scheduled by the gateway are commanded through the downlink and then parsed by the processor node. The processor

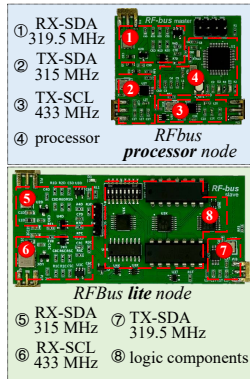


Figure 9: Prototypes.

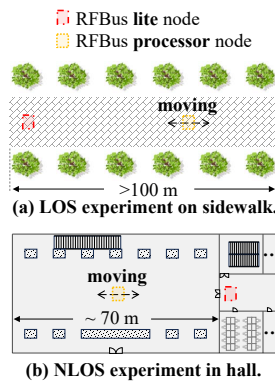


Figure 10: RFBUS Comm.

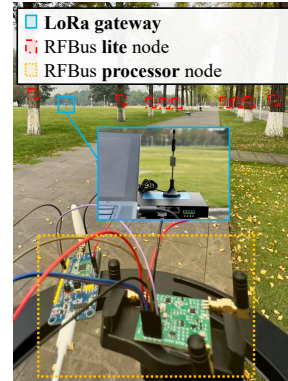


Figure 11: Outdoor Eva.

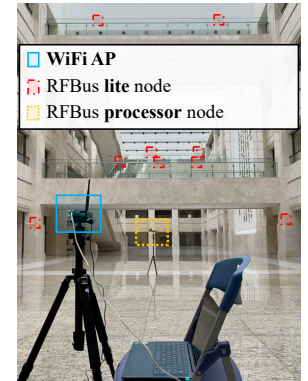


Figure 12: Indoor Eva.

polls the inter-node sensor chips on the lite node(s) for centralized data acquisition. After each chip polling round, the centralized processor node aggregates and packages the obtained sensor data into a frame and then uploads it to the gateway through its on-node COTS RF transceiver.

Such inter-node chip polling is achieved through the RFBUS network, where the processor node behaves as an RFBUS master, initiating RFBUS sessions to access the inter-node sensor chips on the lite nodes. While the lite node behaves as an RFBUS slave, where its on-node sensor chip responds to the inter-node processor on the processor node. Three sets of RFBUS interfaces are used to piggyback I²C signals via three different RF chains/channels. Specifically, SCL generated by the processor is transmitted to the lite node through a set of RFBUS interfaces. At the same time, two sets of RFBUS interfaces are implemented symmetrically to exchange bus data, between local I²C bus on the processor and lite nodes in bi-direction.

6 IMPLEMENTATION

The prototypes are implemented by discrete COTS components. Specifically, the prototypes of processor nodes with mainstream embedded processor STM32F103C6 and RFBUS interface are implemented in two versions. One version equips a LoRa transceiver for outdoor evaluation, and the other one is embedded with a WiFi transceiver for indoor evaluation. The prototype of the lite node is minimalist with the RFBUS interface and the RFBUS frame decoder.

RFBUS interface. RFBUS interface consists of an RF transmitter IC (MAX41461/2) with program-free functionality to present ‘data-in, antenna-out’ modulation and an ultra-low-power demodulator implemented inexpensively using the following components. First, SAW filters with narrow passband support the RFBUS network to simultaneously receive RF carriers that piggyback SDA and SCL signals by frequency-division multiplexing. Further, a low-cost RF Schottky diode (SMS7630) is used to demodulate the received RF carrier by passive envelope detection. An ultra-low-power OA (TLV9001) is used to amplify the analog-basedband, improving receiving sensitivity. Finally, the baseband is normalized into digital format by a voltage comparator (NCS2202SN1T1G), and a digital inverter with open-drain output (SN74LVC1G38) is selected to invert the high voltage-level signal that is detected from the RF carrier to low voltage-level signal to hold local I²C bus.

RFBUS frame decoder. RFBUS frame decoder is implemented on the lite node with five logic components. It parses the 2-byte RFBUS frame head and then activates/shuts down the RFBUS interface (TX) accordingly. A 16-bit shift register (SN76HC164) is used to output received frame data in pipeline. A counter (CD74HC40103) is activated and starts counting down when the RFBUS interface address that is manually configured on the DIP switch is matched. The address decoder is implemented by two serial-connected XNOR gates (CD4077BE). Finally, D-Latch with open-drain output (CD4042BM) outputs a control signal to connect/disconnect the RFBUS interface (TX) with local I²C bus.

In addition, a bi-directional SPDT switch (74LVC1G3157) is embedded at the processor node for half-duplex RF signaling control.

7 EVALUATION

The evaluation is made in two steps.

Step 1: We evaluate the communication range and throughputs of the RFBUS network to clarify the application boundary of the proposed processor-sharing IoT architecture.

Link setup (communication between the processor and lite nodes). The processor and lite nodes communicate through the RFBUS interface that consists of a non-coherent demodulator for RF carrier detection with an antenna gain of 6 dBi and a modulator for RF carrier generation with +17 dBm output power through an antenna gain of +2 dBi.

Step 2: We evaluate the node’s power consumption, manufacturing cost, and task delay in two case studies.

Benchmark selection. The proposed processor-sharing IoT architecture consolidates multiple processor and sensor chips into a sensor cluster with centralized chip polling and aggregated data upload. Given this, we consider conventional embedded architecture the benchmark, where all sensor nodes are full-fledged to read their on-node sensor chips and then forward sensor results to the gateway independently. The conventional node and RFBUS processor node are embedded with the same mainstream processor, STM32F103, while the RFBUS lite node is simplified without a processor.

Link setup (communication between the processor node and the gateway). The processor node uploads aggregated sensor data to the remote gateway through two different kinds of COTS RF transceivers,

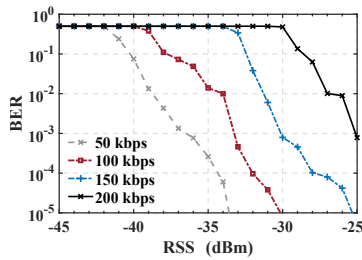


Figure 13: Sensitivity of the envelope detector.

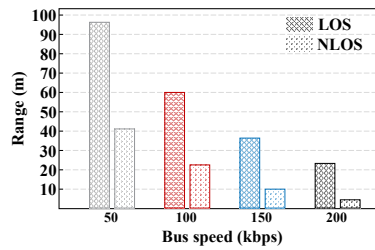


Figure 14: LOS and NLOS inter-node communication ranges.

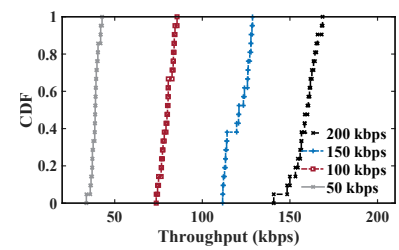


Figure 15: Network throughput.

respectively. Specifically, the WiFi transceiver CC3130 has +17 dBm transmitting power, and the LoRa transceiver SX1276 has +17 dBm transmitting power. Likewise, the benchmark, *i.e.*, conventional full-fledged sensor node, communicates with the gateway through the identical wireless communication setups of WiFi and LoRa for fairness.

7.1 Evaluating RFBUS Network

Inter-node communication range. As IoT sensor nodes are typically deployed at fixed locations, the inter-node communication range of the RFBUS network determines the sensor coverage of a processor node. Given that the RF front-end deployed on the processor and lite nodes are the same, their uplink and downlink communication ranges are identical. We first evaluate the receiving sensitivity of the RF front-end, *i.e.*, the envelope detector, by conducting laboratory tests. We connect the USRP-2922 and the RF signal receiving part of the front-end with RF cable and employ one or two 30dB RF attenuators in the cable. The received signal strength (RSS) is precisely controlled by setting the transmitting signal strength and the number of employed RF attenuators. We measure the bit error rate (BER) for each round by operating the USRP to send predetermined test data over different data rates, *i.e.*, bus speeds. The test data incorporates a group of 8-bit stream '11000101' in a loop 50,000 times to simulate all possible bit-flips, *i.e.*, continuous and alternate bits for both '0' and '1'. The sensitivity of the envelope detector is depicted in Fig. 13 with measured RSS and corresponding BER. We then conduct field trials in both line-of-sight (LOS) and non-line-of-sight (NLOS) environments, as shown in Fig. 10, where a processor node transfers the predetermined test data to a lite node over different bus speeds. We consider the communication range the most extended inter-node data transfer range with a BER lower than 1%. As shown in Fig. 14, the communication range becomes shorter as bus speed increases. The reason is that the envelope detector discriminates between bit '1' and '0' by comparing the signal energy accumulated in a capacitance with the predefined voltage threshold. When the bus speed is increased, the duration of each bit becomes shorter. So, the difference in accumulated voltage levels becomes smaller and thus more prone to error.

Network and task throughputs. We consider a simple RFBUS network that contains one processor node and nine lite nodes. Each node is embedded with three sensor chips, which are temperature sensor [14], photometric sensor [15], and accelerometer [16]. The

configuration time of each sensor chip is 27 ms on average. In each RFBUS session, the processor node randomly selects a lite node, and then reads its on-node sensor chips in two different strategies - polling all three sensor chips in pipeline, or querying sensor chips once a session. The length of the RFBUS frame head and average sensor data, *i.e.*, I^2C payload, are 16 bits and 56 bits, respectively.

The following throughput is measured over different bus speeds. As shown in Fig. 15, bus speed is the bottleneck of the network throughput. Although the latest I^2C bus interface supports bus speeds up to 1 Mbps, the maximum bus speed of the RFBUS interface is around 200 kbps because of the limited conducting frequency of the RF Schottky diode used for envelope detection; however, when we pay attention to the task throughput, as shown in Tbl. 1, the processor node reads these 30 sensor chips up to 826 times in pipeline within a second. It equals to polling 24,780 inter-node sensor chips at a 1 Hz sampling frequency. It is sufficient to support the scenario where a vast number of sensor chips are densely deployed in a small area. In addition, compared with the strategy of querying only one sensor chip once a session, the RFBUS network obtains higher throughput when sensor chips on the same lite node are traversed continuously. This implies that the processor should be programmed to read the sensor chips continuously on the same lite node if it is desired to be shared in a dense sensor deployment scenario.

7.2 Evaluation in Two Case Studies

We demonstrate nodes' deployment costs, *i.e.*, manufacturing cost and power consumption, and response time in conventional and proposed architecture using two case studies. Specifically, the conventional and proposed nodes are configured to measure temperature at ten different outdoor (see Fig. 11) and indoor (see Fig. 12) locations, respectively. It is worth clarifying that, in conventional architecture, a WiFi access point (AP) or LoRa gateway communicates with ten full-fledged embedded sensor nodes simultaneously. In contrast, the processor node first queries the rest of the nine lite nodes in pipeline, and then forwards the aggregated sensor data to a WiFi AP or LoRa gateway uniformly.

Power consumption. Fig. 16 illustrates the breakdown of overall energy consumption at RFBUS sensor nodes, which shows $6.09 \times$ and $6.69 \times$ decrease compared with the conventional WiFi and LoRa sensor nodes, respectively. The main reason is that RFBUS only utilizes the power-hungry WiFi or LoRa transceiver on the processor node to transmit aggregated sensor data but collects raw

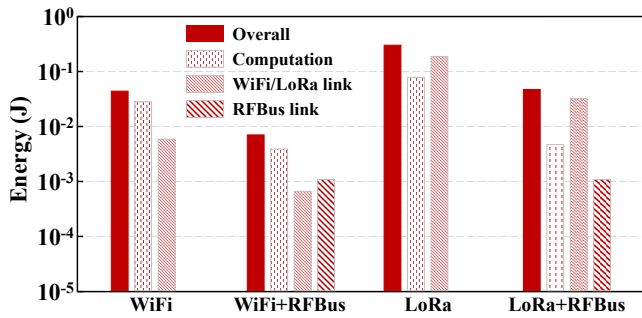


Figure 16: Overall energy consumption of ten sensor nodes are measured in two case studies, where they read temperature for one time at each node and then upload temperature data to WiFi AP or LoRa gateway.

	50 kbps	100 kbps	150 kbps	200 kbps
Strategy	polling	polling	polling	polling
Task throughput	218	403	550	826
Strategy	querying	querying	querying	querying
Task throughput	169	339	452	678

Table 1: Task throughput of RFBUS network measured under different bus speeds and strategies.

sensor data from lite nodes through a low-power RFBUS interface. In comparison, conventional architecture establishes ten power-hungry RF links between the gateway and the full-fledged sensor nodes, where 98.5% and 93.3% of transmitting energy are paid for RF link configuration and maintenance, respectively. In detail, the RFBUS interface only consumes $120 \mu\text{J}$ to transfer 2-byte sensor data from a lite node to a processor node, which is $5.1 \times$ and $160.1 \times$ less than the WiFi and LoRa, respectively. In addition, RFBUS uses a single processor to query the sensor chips deployed at nine different locations instead of running ten processors independently, reducing the static power consumption by 85.1% and 93.2% compared with that consumed on ten full-fledged nodes.

Response time. Response time is an important performance indicator of real-time sensing capability [17], which is measured from the task beginning to the moment when the gateway has received all temperature results from 10 different locations. To make the evaluation convincing, we operate the COTS RF transceivers under their standard data rate, *i.e.*, 1 Mbps for WiFi, 976.5625 symbols per second for LoRa (SF=7, bandwidth=125kHz). At the same time, the RFBUS speed is limited to 100 kbps, which is the bus speed of the I²C standard mode. As illustrated in Tbl. 2, the processor-sharing IoT architecture consumes $0.52 \times$ and $2.43 \times$ shorter response time than conventional architecture adapted to WiFi or LoRa communication, respectively.

When only one sensor node is deployed in field, the response times of the two architectures are identical. Because the processor node can directly access its on-node temperature sensor chip and forward the temperature data to the gateway. In this case, the processor is not shared among its neighboring lite nodes. However, once the amount of deployed sensor nodes is larger than one, *e.g.*,

WiFi		LoRa	
conventional architecture	processor-sharing architecture	conventional architecture	processor-sharing architecture
364 ms	241 ms	1029 ms	300 ms

Table 2: Response time measured in two case studies, where ten sensor nodes read temperature sensor and then upload the sensor data to WiFi AP or LoRa gateway.

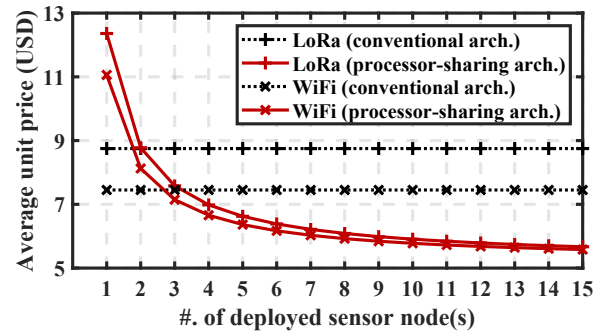


Figure 17: The average unit price of the conventional and the processor-sharing sensor nodes is calculated in two case studies, where nodes communicate with WiFi AP and LoRa gateway, respectively.

ten sensor nodes deployed in these case studies, the processor node accesses the inter-node sensor chips on lite nodes. Then, it forwards the aggregated sensor result to the gateway uniformly. The aggregated payload makes the WiFi and LoRa packet more efficient with a similar frame aggregation functionality, resulting in lower transmission delay. Meanwhile, given that the RFBUS frame structure is more streamlined than that of WiFi and LoRa, the communication overhead of the RFBUS network is much smaller. In summary, the lower transmission delay benefits the proposed architecture by allowing it to achieve better real-time performance. As the scale of IoT deployment increases, this advantage becomes more apparent.

Manufacturing cost. Processor-sharing IoT architecture introduces a noticeable manufacturing cost reduction as processors and COTS RF transceivers are removed from the lite nodes. To make the evaluation referential, we calculate the unit price of the sensor nodes in both discrete and integrated implementations.

Discrete prototypes. Tbl. 3 lists the manufacturing cost of the conventional full-fledged and RFBUS sensor nodes implemented by discrete COTS components. The lite node is much cheaper than the conventional full-fledged ones because it is highly simplified without a processor or COTS RF transceiver embedded. In trade-off, the processor node is more expensive than the conventional full-fledged one because it has additional RFBUS interfaces for RFBUS communication. However, as illustrated in Fig. 17, the manufacturing cost of the processor node is diluted with the increasing amount of lite nodes, resulting in a lower average unit price of the RFBUS sensor node. For instance, when nine lite nodes share a processor node, the average unit price is reduced by 23.5% and 33.5%, respectively. More significant cost reduction is achieved with the combination

	Full-fledged sensor node (WiFi version)	Full-fledged sensor node (LoRa version)	RFBUS processor node (WiFi version)	RFBUS processor node (LoRa version)	RFBUS lite node
Sensor	0.25 USD (1*TMP1075)				
Processor	3.3 USD (1*STM32F103)				0 USD (implemented without processor)
Radio	3.9 USD	5.2 USD	3.9 USD (1*CC3130)	5.2 USD (1*SX1276)	4.94 USD (1*B39431B3710U410+1*RF1432C+2*SMS7630+2*TLV9001+2*NCS2200+4*SN74LVC1G38+1*MAX41461+2*SN74HC164+2*CD4077BE+1*SN74AUP1T87DCKR+1*SN74HCS21PWR+1*NC7SZ175P6X+1*CD74HC40103M96)
	(1*CC3130)	(1*SX1276)	3.61 USD (1*B39321B3722U410+1*SMS7630+1*TLV9001+1*NCS2200+3*SN74LVC1G38+1*SN74LVC1G3157+2*MAX41461)		
TOTAL	7.45 USD	8.75 USD	11.06 USD	12.36 USD	5.19 USD

Table 3: The cost of manufacturing the conventional and processor-sharing sensor nodes by discrete COTS components is calculated in two cases, where nodes communicate with WiFi AP and LoRa gateway, respectively. The unit prices of these components are obtained from the Digikey® e-shop based on the purchase quantity of 3,000 pieces.

of RFBUS and LoRa because the LoRa transceiver is more expensive than the WiFi transceiver. It indicates that the RFBUS network architecture is more cost-efficient in outdoor, large-scale sensor deployment scenarios.

Integrated prototype. Current IoT industry prefers single-chip solution, where the RF transceiver and processor are integrated into a single chip, e.g., ESP32. Compared with the single-chip solution, the cost benefits of the proposed architecture still stand. Although we cannot accurately estimate the unit price of the RFBUS processor and lite nodes after integration because it relates to various factors, e.g., shipment volume, the RFBUS lite node only consumes five essential logic components to realize its local logic operation. Compared with the full-fledged sensor nodes embedded with general-purpose computing units, the computing resources consumed by RFBUS sensor nodes are nearly negligible. Meanwhile, only a few inexpensive components, e.g., SAW, RF diode, comparator, are demanded for RFBUS communication.

8 RELATED WORK

Significant effort has been made in the literature to relieve the burden of large-scale IoT deployment.

Computation offload. Some effort [5–8, 18, 19] tries to relieve the computation burden of large-scale IoT deployment by shifting some of the on-node computation workloads, e.g., sampling and data acquisition, from distributed sensor nodes to a centralized gateway. To offload these computation workloads, sensor nodes must communicate with the gateway more frequently. The communication overhead becomes significant, so these sensor nodes can only work with ultra-low-power but short-range RF communication schemes, e.g., the passive communication within 30 m. Otherwise, the cost of computation offload, i.e., more communication overhead, will outweigh the benefit, i.e., less local computation workload. The reduced network scalability cannot allow large-scale IoT deployment unless expensive power-hungry gateways are densely deployed in trade-off, increasing overheads at the system level. In contrast, our proposed architecture leverages the processor on a few full-fledged processor nodes to access their neighboring sensor chips via RFBUS with negligible overheads. At a high level, it is also a kind of computation offload design. However, the major difference is that the computation workload of the lite sensor nodes is shifted to the neighboring full-fledged sensor nodes rather than the remote gateway. By embedding only five logic components at each lite node, RFBUS constructs a multi-to-multi chip-oriented RF network between processor and sensor chips that supports addressing towards

2,048 chips with anti-collision and reliable delivery functionality. Moreover, part of the computation workload of the gateway is also shifted to the processor nodes, as the gateway only needs to schedule a small number of processor nodes. Gateway’s larger-scale sensor networking potential is inspired; thus, the practical network scalability is enlarged rather than reduced.

RF power reduction. The power reduction of RF transceivers often plays an important role in nodes’ power optimization. Some effort [20–27] improves the energy efficiency of long-range communication means, e.g., LoRa and NB-IoT, by modifying their link layer protocol and transmission mechanism. However, they can hardly touch the essence of low-power demand because the instantaneous transmission and reception power consumption are still high. E-MiLi [28] addresses the energy wastage due to WiFi carrier detection, while SlimWiFi [29] realizes the power reduction of WiFi transmitter. For the emerging millimeter-wave communication, mmPlug [30] reduces its power consumption through the positive feedback effect of amplification circuits. Further, plenty efforts [2, 31–34] adopt the node device to passive communication scheme in WiFi [35], LoRa [4, 36–39], BLE [40], LiFi [41], mm-wave [42] and underwater [43] communication systems. Although they reduce RF communication power consumption to the μ W level, the communication range and network scalability, however, are sacrificed. Thus, more IoT gateways need to be deployed for large-scale IoT deployment, which increases the deployment costs at the system level. In contrast, the processor-sharing architecture designs a low-power inter-node RF communication interface to enable processors to access their neighboring sensor chips through streamlined but powerful I²C protocol with negligible overheads, and then leverage the COTS RF transceiver on the processor to upload the aggregated sensor data to the remote gateway. The communication overheads are reduced in both physical and link layers.

Multi-hop wireless sensor network. Multi-hop networks [44–49] are widely researched in the networked sensor system, where sensor data obtained by a wireless sensor network (WSN) node is relayed to a remote gateway with the hop-by-hop scheme. Nodes in a multi-hop network must be full-fledged embedded computers since they must accomplish sensor chip access control locally and establish dynamic multi-hop communication links with other nodes. Consequently, multi-hop networks consume significant communication overheads for maintenance, such as updating the routing table and configuring dynamic links. Thus, their valuable network throughput is wasted in data relay operation and further decreased

as the number of multi-hop nodes increases, resulting in poor robustness and significant communication latency [50]. In contrast, in our proposed architecture, the processor and lite nodes are organized in a static star topology. The communication protocol is executed by the I^2C bus interface, and the frames are designed to be minimalist. Therefore, a lite node does not need to be equipped with a general-purpose computing unit to perform sensor chip access control and communication protocol stacks locally, nor does it need to be equipped with a power-hungry RF transceiver for long-range data transmission. Instead, their sensor data is aggregated at the processor node and uniformly uploaded to a remote gateway through another well-established star-topology network.

High-efficient computation. Also, designing a processor that fits the workload of an IoT sensor node is a promising research area. There has been abundant research on cutting-edge computer architecture that aims for inexpensive, long-lasting, and standalone sensor nodes coupled with low computation workload. Hempstead [51] uncovers the inefficient processor usage, and the following Snafu [52] develops an ultra-low-power architecture to provide suitable computation resources for various nodes flexibly. While processor design is also wildly studied to meet the resource-sensitive computation demands, such as bespoke IP [53], manufacturing technique [54, 55], fine-grained power management [56, 57] and code generation [58–60]. Motivated by these efforts in the computer architecture community, we share the underutilized processor on full-fledged sensor nodes with neighboring sensor chips for better computation efficiency and overall cost reduction.

9 DISCUSSION

There are some design concerns and limitations in the current prototypes, as discussed below.

Workload increased at the processor nodes. The lite nodes are minimalist; the processor nodes, however, are more energy-consuming than conventional full-fledged sensor nodes as they access the inter-node sensor chips on these lite nodes. Such an issue, however, is not significant since the additional RFBUS interface is designed to operate with low power consumption, and the RFBUS frame is streamlined. On the other hand, RFBUS is a multi-to-multi network; thus, the gateway can balance the workload among processor nodes to prevent the energy from being consumed too quickly at a particular processor node. In order to reduce maintenance effort at the system level, it is acceptable to replace the batteries of a few processor nodes or to equip energy harvesting components, e.g., solar panel, for them.

Single-point failure. A disabled processor node causes serious run-time troubles as the lite nodes are functionalized by it. Luckily, RFBUS is a multiple-to-multiple network where multiple processor nodes can co-exist. The gateway can periodically query the processor nodes and dispatch a neighboring processor node to replace the disabled one.

Security and privacy. Security and privacy are indispensable for IoT systems [61–64]. Otherwise, a burglar may pick homes to rob by accessing Heating Ventilation Air Conditioning (HVAC) sensors to find vacant homes. Also, it is potential for an attacker to interfere with nefarious purposes rather than just eavesdrop. In the iteration of the proposed architecture, RFBUS sensor nodes can

be secured by embedding a specialized encryption circuit that is backward compatible with bus specifications.

Bus compatibility beyond I^2C bus. The I^2C -version RFBUS is an early step of the proposed processor-sharing paradigm. The main reason for choosing I^2C lies in its well-established link layer protocol executed by each I^2C bus interface; thus, the computation burden of link layer control can be released from the lite nodes. Besides standard I^2C , current RFBUS is also backward compatible with multiple I^2C -varieties, such as SMBUS and the latest I^3C , as they share the exact PHY layer specification. Meanwhile, there are some buses that might be good choices for RFBUS. For instance, the CAN bus is widely adopted for industrial sensor chips. If the CAN interface can somehow be adapted to the RFBUS network, the processor-sharing paradigm will shine in more industrial scenarios.

10 CONCLUSION

We present a novel processor-sharing IoT architecture for large-scale sensor deployment, which achieves $6.09 \times$ and $6.69 \times$ energy saving in outdoor and indoor scenarios, respectively, and reduces the unit price of sensor nodes by 23.5% and 33.5%. The majority of the proposed sensor nodes are highly simplified to operate as RF peripherals rather than full-fledged embedded computers. Their on-node sensor chips are read by their neighboring full-fledged processor node with negligible overheads via RFBUS. On the one hand, the RFBUS interface is backward compatible with the I^2C bus specification; thus, RFBUS sensor nodes inherit versatile link layer services transparently from the I^2C bus. On the other hand, the RFBUS interface is designed with joint considerations of system-level performance and deployment costs, inspiring the large-scale deployment potential for the networked sensors.

ACKNOWLEDGMENTS

We sincerely thank our anonymous shepherd and all reviewers for their time and effort in reviewing our work and providing valuable feedback. This work was supported by the National Natural Science Foundation of China under Grant U21A20462 and 62432008.

REFERENCES

- [1] The New York Times. Death toll of maui wildfire rises to 101. <https://www.nytimes.com/article/maui-wildfire-victims.html>, 2024. Accessed: 2024-02-13.
- [2] Mohammad Rostami, Karthik Sundaresan, Eugene Chai, Sampath Rangarajan, and Deepak Ganesan. Redefining passive in backscattering with commodity devices. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.
- [3] Mehrdad Hesar, Ali Najafi, and Shyamnath Gollakota. NetScatter: Enabling Large-Scale backscatter networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 271–284, 2019.
- [4] Yuxiang Peng, Shiyue He, Yu Zhang, Zhiang Niu, Lixia Xiao, and Tao Jiang. Ambient lora backscatter system with chirp interval modulation. *IEEE Transactions on Wireless Communications*, 22(2):1328–1342, 2022.
- [5] Songfan Li, Chong Zhang, Yihang Song, Hui Zheng, Lu Liu, Li Lu, and Mo Li. Internet-of-microchips: direct radio-to-bus communication with spi backscatter. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.
- [6] Songfan Li, Qianhe Meng, Yanxu Bai, Chong Zhang, Yihang Song, Shengyu Li, and Li Lu. Go beyond rfid: Rethinking the design of rfid sensor tags for versatile applications. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, ACM MobiCom '23*, New York, NY, USA, 2023. Association for Computing Machinery.
- [7] Chong Zhang, Songfan Li, Yihang Song, Qianhe Meng, Minghua Chen, Yanxu Bai, Li Lu, and Hongzi Zhu. Lego: Empowering chip-level functionality plug-and-play for next-generation iot devices. In *Proceedings of the 28th ACM International*

- Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, page 404–418, 2023.
- [8] Yihang Song, Chao Song, Li Lu, Shen Yang, Songfan Li, Chong Zhang, Qianhe Meng, Xiandong Shao, and Haili Wang. Chipnet: Enabling large-scale backscatter network with processor-free devices. *ACM Transactions on Sensor Networks*, 18(4):1–26, 2022.
 - [9] Mohamad Katanbaf, Anthony Weinand, and Vamsi Talla. Simplifying backscatter deployment: Full-Duplex LoRa backscatter. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 955–972. USENIX Association, April 2021.
 - [10] Dinesh Bharadia, Kiran Joshi, and Sachin Katti. Robust full duplex radio link. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, page 147–148, New York, NY, USA, 2014. Association for Computing Machinery.
 - [11] Xiuzhen Guo, Longfei Shangguan, Yuan He, Nan Jing, Jiacheng Zhang, Haotian Jiang, and Yunhao Liu. Saiyan: Design and implementation of a low-power demodulator for LoRa backscatter systems. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 437–451, Renton, WA, April 2022. USENIX Association.
 - [12] Yihang Song, Li Lu, Jiliang Wang, Chong Zhang, Hui Zheng, Shen Yang, Jinsong Han, and Jian Li. μ Mote: Enabling passive chirp de-spreading and μ W-level Long-Range downlink for backscatter devices. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1751–1766, Boston, MA, April 2023. USENIX Association.
 - [13] Songfan Li, Hui Zheng, Chong Zhang, Yihang Song, Shen Yang, Minghua Chen, Li Lu, and Mo Li. Passive DSSS: Empowering the downlink communication for backscatter systems. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 913–928, Renton, WA, April 2022. USENIX Association.
 - [14] TEXAS INSTRUMENTS. Temperature sensor tmp1075. <https://www.ti.com/lit/ds/symlink/tmp1075.pdf>.
 - [15] Analog Devices. Photometric sensor adux1020. <https://www.analog.com/media/en/technical-documentation/data-sheets/ADUX1020.pdf>.
 - [16] NXP Semiconductors. 6-axis accelerometer sensor fxos8700cq. <https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/609/FXOS8700CQ.pdf>.
 - [17] Michael Spörk, Carlo Alberto Boano, and Kay Römer. Improving the timeliness of bluetooth low energy in dynamic rf environments. *ACM Trans. Internet Things*, 1(2), April 2020.
 - [18] Pengyu Zhang, Pan Hu, Vijay Pasikanti, and Deepak Ganesan. Ekhneth: High speed ultra low-power backscatter for next generation sensors. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 557–568. ACM, 2014.
 - [19] Sicong Liu, Bin Guo, Cheng Fang, Ziqi Wang, Shiyan Luo, Zimu Zhou, and Zhiwen Yu. Enabling resource-efficient aiot system with cross-level optimization: A survey. *IEEE Communications Surveys and Tutorials*, 26(1):389–427, 2024.
 - [20] Zehua Sun, Tao Ni, Huanqi Yang, Kai Liu, Yu Zhang, Tao Gu, and Weitao Xu. Flora: Energy-efficient, reliable, and beamforming-assisted over-the-air firmware update in lora networks. In *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, pages 14–26, 2023.
 - [21] Yinghui Li, Jing Yang, and Jiliang Wang. Dylora: Towards energy efficient dynamic lora transmission control. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2312–2320. IEEE, 2020.
 - [22] Fu Yu, Xiaolong Zheng, Liang Liu, and Huadong Ma. Enabling concurrency for non-orthogonal lora channels. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.
 - [23] Deliang Yang, Xianghui Zhang, Xuan Huang, Liqian Shen, Jun Huang, Xiangmao Chang, and Guoliang Xing. Understanding power consumption of nb-iot in the wild: Tool and large-scale measurement. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, MobiCom '20*, New York, NY, USA, 2020. Association for Computing Machinery.
 - [24] Akshay Gadre, Revathy Narayanan, Anh Luong, Anthony Rowe, Bob Iannucci, and Swarun Kumar. Frequency configuration for low-power wide-area networks in a heartbeat. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 339–352, 2020.
 - [25] Absar-Ul-Haque Ahmar, Emekcan Aras, Thien Duc Nguyen, Sam Michiels, Wouter Joosen, and Danny Hughes. Design of a robust mac protocol for lora. *ACM Trans. Internet Things*, 4(1), February 2023.
 - [26] Roman Trüb, Reto Da Forno, Andreas Biri, Jan Beutel, and Lothar Thiele. Lsr: Energy-efficient multi-modulation communication for inhomogeneous wireless iot networks. *ACM Trans. Internet Things*, 4(2), April 2023.
 - [27] Emekcan Aras, Stéphane Delbruel, Fan Yang, Wouter Joosen, and Danny Hughes. Chimera: A low-power reconfigurable platform for internet of things. *ACM Trans. Internet Things*, 2(2), March 2021.
 - [28] Xinyu Zhang and Kang G Shin. E-mili: Energy-minimizing idle listening in wireless networks. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 205–216, 2011.
 - [29] Renjie Zhao, Kejia Wang, Kai Zheng, Xinyu Zhang, and Vincent Leung. SlimWiFi: Ultra-Low-Power IoT radio architecture enabled by asymmetric communication. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1201–1219, 2023.
 - [30] Mohammad Mazaheri, Rafael Ruiz, Domenico Giustiniano, Joerg Widmer, and Omid Abari. Bringing millimeter wave technology to any iot device. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.
 - [31] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R Smith. Ambient backscatter: Wireless communication out of thin air. *ACM SIGCOMM computer communication review*, 43(4):39–50, 2013.
 - [32] Aaron N Parks, Angli Liu, Shyamnath Gollakota, and Joshua R Smith. Turbocharging ambient backscatter communication. *ACM SIGCOMM Computer Communication Review*, 44(4):619–630, 2014.
 - [33] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 356–369, 2016.
 - [34] Mauro Piva, Andrea Coletta, Gaia Maselli, and John A. Stankovic. Environment-driven communication in battery-free smart buildings. *ACM Trans. Internet Things*, 2(2), April 2021.
 - [35] Dinesh Bharadia, Kiran Raj Joshi, Manikanta Kotaru, and Sachin Katti. Backfi: High throughput wifi backscatter. *ACM SIGCOMM Computer Communication Review*, 45(4):283–296, 2015.
 - [36] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. Plora: A passive long-range data network from ambient lora transmissions. In *Proceedings of the 2018 conference of the ACM special interest group on data communication*, pages 147–160, 2018.
 - [37] Vamsi Talla, Mehrdad Hesar, Bryce Kellogg, Ali Najafi, Joshua R Smith, and Shyamnath Gollakota. Lora backscatter: Enabling the vision of ubiquitous connectivity. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 1(3):1–24, 2017.
 - [38] Jinyan Jiang, Zhenqiang Xu, Fan Dang, and Jiliang Wang. Long-range ambient lora backscatter with parallel decoding. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 684–696, 2021.
 - [39] Yidong Ren, Puyu Cai, Jinyan Jiang, Jialuo Du, and Zhichao Cao. Prism: High-throughput lora backscatter with non-linear chirps. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.
 - [40] Yuri Murillo, Alessandro Chiumento, Brecht Reynders, and Sofie Pollin. An all-wireless sdn framework for ble mesh. *ACM Trans. Internet Things*, 1(4), August 2020.
 - [41] Muhammad Sarmad Mir, Borja Genoves Guzman, Ambuj Varshney, and Domenico Giustiniano. Passivelifi: Rethinking lora for low-power and long range rf backscatter. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 697–709, 2021.
 - [42] Haofan Lu, Mohammad Mazaheri, Reza Rezvani, and Omid Abari. A millimeter wave backscatter network for two-way communication and localization. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 49–61, 2023.
 - [43] Aline Eid, Jack Rademacher, Waleed Akbar, Purui Wang, Ahmed Allam, and Fadel Adib. Enabling long-range underwater backscatter via van Atta acoustic networks. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 1–19, 2023.
 - [44] Yung Yi and Sanjay Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. *IEEE/ACM Transactions on Networking*, 15(1):133–144, 2007.
 - [45] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '04*, page 133–144, New York, NY, USA, 2004. Association for Computing Machinery.
 - [46] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. *MobiCom '03*, page 66–80, New York, NY, USA, 2003. Association for Computing Machinery.
 - [47] Y. Thomas Hou, Yi Shi, and Hanif D. Serali. Spectrum sharing for multi-hop networking with cognitive radios. *IEEE Journal on Selected Areas in Communications*, 26(1):146–155, 2008.
 - [48] Shamik Sengupta and K.P. Subbalakshmi. Open research issues in multi-hop cognitive radio networks. *IEEE Communications Magazine*, 51(4):168–176, 2013.
 - [49] Beshr Al Nahas, Antonio Escobar-Molero, Jirka Klauke, Simon Duquenooy, and Olaf Landsiedel. Blueflood: Concurrent transmissions for multi-hop bluetooth 5—modeling and evaluation. *ACM Trans. Internet Things*, 2(4), July 2021.
 - [50] Geir Egeland and Paal E Engelstad. The availability and reliability of wireless multi-hop networks with stochastic link failures. *IEEE Journal on Selected Areas in Communications*, 27(7):1132–1146, 2009.
 - [51] Mark Hempstead, Nikhil Tripathi, Patrick Mauro, Gu-Yeon Wei, and David Brooks. An ultra low power system architecture for sensor network applications. In *32nd International Symposium on Computer Architecture (ISCA)*, pages 208–219. IEEE, 2005.

- [52] Graham Gobieski, Ahmet Oguz Atli, Kenneth Mai, Brandon Lucia, and Nathan Beckmann. Snafu: an ultra-low-power, energy-minimal cgra-generation framework and architecture. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 1027–1040. IEEE, 2021.
- [53] Hari Cherupalli, Henry Duwe, Weidong Ye, Rakesh Kumar, and John Sartori. Bespoke processors for applications with ultra-low area and power constraints. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 41–54, 2017.
- [54] Antonio Pullini, Davide Rossi, Igor Loi, Giuseppe Tagliavini, and Luca Benini. Mr.wolf: An energy-precision scalable parallel ultra low power soc for iot edge processing. *IEEE Journal of Solid-State Circuits*, 54(7):1970–1981, 2019.
- [55] Christopher Celio, Pi-Feng Chiu, Krste Asanović, Borivoje Nikolić, and David Patterson. Broom: An open-source out-of-order processor with resilient low-voltage operation in 28-nm cmos. *IEEE Micro*, 39(2):52–60, 2019.
- [56] Jawad Haj-Yahya, Mohammed Alser, Jeremie Kim, A Giray Yağlıkcı, Nandita Vijaykumar, Efraim Rotem, and Onur Mutlu. Sysscale: Exploiting multi-domain dynamic voltage and frequency scaling for energy efficient mobile processors. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 227–240. IEEE, 2020.
- [57] Leyla Nazhandali, Bo Zhai, A Olson, Anna Reeves, Michael Minuth, Ryan Helfand, Sanjay Pant, Todd Austin, and David Blaauw. Energy optimization of subthreshold-voltage sensor network processors. In *32nd International Symposium on Computer Architecture (ISCA'05)*, pages 197–207. IEEE, 2005.
- [58] Victor A Ying, Mark C Jeffrey, and Daniel Sanchez. T4: Compiling sequential code for effective speculative parallelization in hardware. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 159–172. IEEE, 2020.
- [59] Qun Huang, Siyuan Sheng, Xiang Chen, Yungang Bao, Rui Zhang, Yanwei Xu, and Gong Zhang. Toward nearly-zero-error sketching via compressive sensing. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 1027–1044, 2021.
- [60] Fan Lai, Jie You, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Sol: Fast distributed computation over slow networks. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 273–288, 2020.
- [61] Nada Alhirabi, Omer Rana, and Charith Perera. Security and privacy requirements for the internet of things: A survey. *ACM Trans. Internet Things*, 2(1), February 2021.
- [62] Amit Kumar Sikder, Leonardo Babun, Z. Berkay Celik, Hidayet Aksu, Patrick McDaniel, Engin Kirda, and A. Selcuk Uluagac. Who's controlling my device? multi-user multi-device-aware access control system for shared smart home environment. *ACM Trans. Internet Things*, 3(4), September 2022.
- [63] Hanwei Zhu, Sid Chi-Kin Chau, Gladhi Guarddin, and Weifa Liang. Integrating iot-sensing and crowdsensing with privacy: Privacy-preserving hybrid sensing for smart cities. *ACM Trans. Internet Things*, 3(4), September 2022.
- [64] Nada Alhirabi, Omer Rana, and Charith Perera. Security and privacy requirements for the internet of things: A survey. *ACM Trans. Internet Things*, 2(1), February 2021.