

Extending Coverage through Integrating Multiple Low-Power Wide-Area Networks: A Latency Minimizing Approach

Prashant Modekurthy*, Dali Ismail[†], Mahbubur Rahman[‡], Abusayeed Saifullah[§]

*University of Nevada, Las Vegas, NV, USA

[†] Binghamton University, NY, USA

[‡]City University of New York, NY, USA

[§]Wayne State University, MI, USA

Abstract—Industrial and agricultural Internet of Things (IoT) are emerging in very large-scale and wide-area applications (e.g., oil-field management, smart farming) that may spread over hundreds of square miles (e.g., 45mi×12mi East Texas Oil-field). Although a single Low-Power Wide-Area Network (LPWAN) covers several miles, it faces coverage challenge in such extremely large-area IoT applications, especially in rural or remote areas with no/limited infrastructure, requiring an in-band integration of multiple LPWANs. We consider a seamless integration of multiple SNOW LPWANs. SNOW (Sensor Network Over White spaces) is an LPWAN architecture over the TV white spaces, avoiding overcrowding problems in the limited ISM band and the cost of licensed band and infrastructure. It offers high scalability through concurrent and bi-directional communication between a base station and numerous nodes. Existing integration of multiple SNOW LPWANs does not consider minimizing network latency and is less suitable for delay-sensitive or real-time applications. In this work, we propose the *first latency-minimizing scalable in-band integration of multiple SNOWs*. Considering the impact of bandwidth on latency and base station power dissipation, low-latency integration of multiple SNOWs as a constrained spectrum allocation problem is formulated. A novel greedy latency- and traffic- aware spectrum allocation to allocate each link's bandwidth is proposed, achieving an integrated network. To enable low-latency integration, we propose two medium access control protocols for multiple SNOWs, RI-TDMA and TDMA, and estimate their latency. We have implemented the proposed integration both on SNOW hardware and in NS-3 simulator. The physical experiments show up to 44% reduction in the maximum network latency under our approach compared to existing approach. The simulation results show at least 62.3% reduction of maximum network latency by the proposed approach with RI-TDMA and 86.7% with TDMA.

Index Terms—Low Power Wide Area Networks, SNOW LPWAN, LPWAN Integration, Low-Latency

1 INTRODUCTION

As an emerging Internet-of-Things (IoT) technology, Low-Power Wide-Area Network (LPWAN) enables low-power (milliwatts), and low data-rate (kbps) information collection from sensors deployed over long distances (a few miles) using narrowband channels (kHz) and storage on the cloud [1]. With the fast growth of IoT, multiple LPWAN technologies have recently emerged such as LoRa, SigFox, IQRF, RPMA, DASH7, Weightless-N/P, Telensa in the ISM band, and EC-GSM-IoT, NB-IoT, LTE Cat M1 in the licensed cellular band [2]. To avoid the *crowd* in the *limited* ISM band and the *cost* of licensed band, *SNOW* (*Sensor Network Over White spaces*) was proposed as an LPWAN architecture to support scalable wide-area IoT over the TV white spaces [3]. *White spaces* are the allocated but locally unused TV spectrums where unlicensed devices can operate as secondary users [4]. To learn about white spaces at a location, a device needs to either sense the medium before transmitting, or consult with a cloud-hosted geo-location database, either periodically or every time it moves 100 meters [4]. Compared to the ISM band, the TV white spaces have lower frequencies (e.g., 54 – 862kHz in USA) and much wider, less crowded spectrum in both rural and most urban areas, with an abundance in the former [5].

Although a single LPWAN covers an area with a radius

of several miles, it faces coverage and scalability challenges in very large-area (e.g., city-wide) deployment [6], [7], [8], [9]. Today, industrial and agricultural IoT and cyber-physical systems are emerging in large-scale and wide-area applications. Specifically, agricultural fields [10], [11] and oil/gas fields [12] may extend over hundreds of square kms. For example, the East Texas Oil-field extends over an area of 45mi×12mi square miles requiring tens of thousands of sensors for management [13]. Emerson is targeting to deploy 10,000 nodes for managing an oil-field in Texas [14], [15]. To cover such large-areas for agricultural/industrial IoT, we need to integrate multiple LPWANs. LPWANs are usually limited to star topology, and rely mostly on wired infrastructure (e.g., cellular LPWANs) or Internet (e.g., LoRa) to integrate multiple networks to cover large areas. However, lack of infrastructure hinders their adoption to rural/remote area applications such as agricultural/industrial IoT. In this paper, we propose a scalable and low-latency in-band integration of multiple LPWANs under limited infrastructure.

We consider integrating multiple SNOWs whose conceptual notion can also be extended to other LPWANs. SNOW offers high scalability due to concurrent asynchronous and bi-directional communication between a base station (BS) and numerous nodes [3]. Its implementation is available as

open-source [16]. Due to their rapid growth, LPWANs in the ISM band will suffer from crowded spectrum, making it critical to exploit white spaces. Compared to cellular LPWANs, SNOW does not need wired infrastructure making it suitable in both rural and urban areas. Due to abundant white spaces, it is a promising platform for smart farming, a global need and recommendation by the United Nations to meet the 70% more food demand by 2050 [17], [18]. Study shows that smart farming powered by IoT can double the produce at low cost by better measuring soil nutrients, moisture, fertilizer, seeds, and storage temperature through dense sensor deployment [2]. Thus, industries such as Microsoft [19] and Monsanto [20] are promoting agricultural IoT.

In-band integration considered in this paper is conceptually different from and challenging than traditional tiered or clustered wireless sensor network (WSN) [21], [22], [23], wireless sensor networks [24], packet relaying in LPWAN [25], [26] or vehicular networks [27]. Typically, in these traditional networks, every link operates on its predetermined bandwidth, and hence, bandwidth allocation is not a concern. In contrast, in the proposed integration, we must assign bandwidth to each link; these bandwidths among the links can be asymmetric (i.e., of different widths), inter-dependent, and need to be determined optimally or effectively for the sake of network performance, subject to BS power dissipation. Additionally, the unique features of SNOW, including massive parallel communication, require a new approach for a seamless integration that will enable concurrent inter- and intra-SNOW communication.

A seamless integration of SNOWs was studied recently in [28]. The existing integration aims to improve scalability (number of total nodes) while minimizing inter-SNOW interference. To achieve scalability, its bandwidth allocation for inter-SNOW communication does not consider the link traffic. As a result, its network latency can be high, eventually affecting the scale. To demonstrate the high network latency of the existing approach, we evaluated the maximum and average latency of the existing approach through simulations in NS-3 [29]. For the simulation, 5 SNOW networks were placed randomly in a circular area with a radius of 5 km, where the BSs were connected in a tree topology. Nodes within the integrated network transmitted packets periodically, with a period of 64 s, using a CSMA-CA MAC as proposed in [28]. We varied the number of nodes within each SNOW from 200 to 1000. Fig. 1 reports that the maximum and average latency for collecting a packet increases exponentially with the number of nodes. We observed that the collisions within intra-SNOW communication and bandwidth allocation along the tree link contributed to the exponential increase of maximum and average latency. Thus, the existing approach is less suitable for delay-sensitive and real-time applications. Note that many WSN applications are time-sensitive. A recent survey on 311 industries conducted by ON World and the International Society of Automation shows that 57%

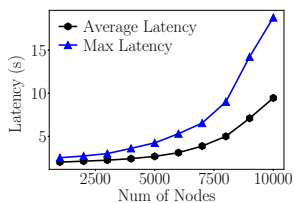


Fig. 1: Latency under Existing SNOW Integration.

of industrial IoT professionals target LPWAN for industrial WSN applications [30].

We propose the first scalable in-band integration of multiple SNOWs that minimizes network latency. In each SNOW, we adopt new media access control (MAC) protocols that blends with the proposed low-latency integration. Our contributions are listed as follows.

- By taking into account the impact of bandwidth on latency and BS power dissipation, we formulate latency minimizing integration of multiple SNOWs as a constrained optimal spectrum allocation problem.
- A greedy algorithm solves the formulated problem by analyzing network latency and adopting a traffic-aware bandwidth allocation along the links to achieve an integrated network. This approach can be extended to future LPWANs in white spaces (e.g., those based on upcoming 802.15.4m standard [31]).
- We have implemented the proposed integration on SNOW hardware platform and observed, through physical experiments, up to 44% decrease in network latency compared to the existing approach [28]. We have also performed simulations in NS-3 [29] and have observed up to 97% decrease in network latency in an integrated network of 12500 nodes (of 25 SNOWs) based on our approach compared to the existing approach.

The paper is organized as follows. Section 2 presents an overview of SNOW. Section 3 gives an overview of the proposed integration along with related work, challenges, and problem formulation. Section 4 details the solution for latency minimizing integration. Section 5, 6, and 7 present experiments, simulations, and conclusion respectively.

2 THE SNOW ARCHITECTURE

SNOW is a promising LPWAN technology with open-source implementation on commercially available off-the-shelf devices [16]. It exploits the widely available TV white spaces for communication, thereby avoiding the crowded ISM band and the cost of licensed band and its associated infrastructure. Its full description is available in [3]. A single SNOW consists of a line-powered and Internet-connected BS and many battery-powered nodes (sensor nodes). Each node is equipped with a single half-duplex radio. To facilitate concurrent bidirectional communication, the BS uses two radios – one only for transmission (called *Tx radio*) and the other only for reception (called *Rx radio*), as shown in Fig. 2. A dual-radio USRP (universal software radio peripheral) connected to Raspberry PI or Laptop is used as the BS. A TI CC1310 device is used as a SNOW node. CC1310 is a tiny, cheap (< \$5), and commercially off-the-shelf (COTS) device with a programmable physical layer [32].

In SNOW, nodes are directly connected to a BS enabling machine-to-machine communication, as shown in Fig. 2. The BS periodically determines white spaces by providing locations of the devices in a cloud-hosted database through the Internet. Since the nodes are power-constrained, they do not determine white spaces. The BS splits a wide white space spectrum into narrowband orthogonal subcarriers (f_1, f_2, \dots), each of equal bandwidth. Note that, **bandwidth**

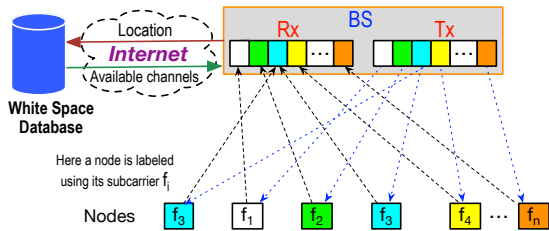


Fig. 2: SNOW architecture with dual radio BS.

of a subcarrier is the width of the spectrum selected for a subcarrier. Upon identifying available subcarriers, the BS assigns one subcarrier to a node for communication, and the node sends/receives on the assigned subcarrier. For a large network, multiple nodes can be assigned a single subcarrier.

To enable many simultaneous transmissions/receptions, the physical layer is designed based on a Distributed implementation of OFDM (Orthogonal Frequency Division Multiplexing), named **D-OFDM**. D-OFDM splits a wide spectrum into many narrowband orthogonal subcarriers enabling parallel data streams to/from numerous nodes from/to the BS. A subcarrier bandwidth is in kHz (e.g., 50kHz, 100kHz), and it varies with packet size and bit rate. The nodes transmit/receive on orthogonal subcarriers. A subcarrier is modulated using Binary Phase Shift Keying (BPSK) or Amplitude Shift Keying (ASK). If the BS spectrum is split into m subcarriers, a BS can receive from m nodes simultaneously on one antenna. Similarly, it can transmit multiple packets on m subcarriers in a single transmission.

The BS uses the same bandwidth and subcarriers for both radios. The subcarriers in the Rx radio are for receiving while those in the Tx radio are for transmitting. SNOW can adopt any MAC protocol that is suitable for the energy-constrained nodes. In its default design, nodes use a lightweight CSMA/CA (carrier sense multiple access with collision avoidance) based MAC protocol for transmission that uses a static interval for random back-off. In this paper, we adopt a time-triggered MAC protocol that blends with the proposed low-latency integration.

3 INTEGRATING MULTIPLE SNOWS: AN OVERVIEW OF THE PROPOSED APPROACH

3.1 System Model and Objective

A WSN is a network of sensors that deliver their data to a base station. It has myriads of applications such as process management, data center management, and monitoring of habitat, environment, volcano, and civil infrastructure [3]. Many WSNs are characterized by a dense and large number of nodes, small packets, low data rate, and low cost. To cover a large area with numerous devices, traditional short range WSN technologies such as IEEE 802.15.4, Bluetooth, and IEEE 802.1 form many-hop mesh networks, thereby limiting the scale. LPWAN is promising technology to overcome this limitation of short range WSN technologies.

Note that a single SNOW (i.e., one BS and its associated nodes) can cover a wide area with a radius of several miles. However, it may not cover an extremely large area such as agricultural farms and oil/gas fields that span hundreds of square miles. To cover such a large area, we need to deploy multiple SNOWs. Therefore, to enable agricultural IoT (in farms) or industrial IoT (in oil/gas field), all of these

deployed SNOWs need to be integrated to facilitate data collection. Note that, in such an integrated network, one Internet-connected BS can collect data from all other BSs. At the other BSs, Internet may not be available. In this work, our objective is to enable scalability and extended coverage in a wide area through a seamless integration of multiple SNOWs which will enable low-latency data collection at one BS from the entire network.

Similar to popular LPWANs such as LoRa, SNOW BSs also have a constant supply of power. This requirement can be met in practice by deploying BSs at existing line power locations such as fences, irrigation machines in agricultural IoT, or oil rigs in Industrial IoT. Furthermore, at locations without a line power, BSs can rely on solar panels connected to batteries for a constant power supply.

In this paper, we consider a network with N BSs, denoted by $BS_0, BS_1, \dots, BS_{N-1}$, where BS_i is the base station of $SNOW_i$, and BS_0 is the *root BS* that collects data from the entire network, i.e., from all SNOWs in the integrated network. BS_0 is connected to the

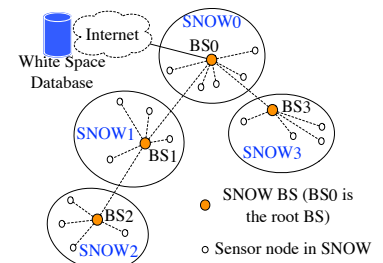


Fig. 3: Integrating Multiple SNOWs to Build a SNOW-tree.

white space database through the Internet. The location of BS_0 is determined based on the Internet availability; for example, in smart agriculture, the root BS is located at a barn or farmer's residence. BS_0 finds white spaces for all SNOWs and allocates spectrum among all SNOWs. We consider that the SNOWs form an in-band SNOW-tree (Fig. 3) like a cluster tree of IEEE 802.15.4m [31]. Note that there can be many ways to construct a tree of the BSs. Varying this topology will vary the performance of the integrated network. However, we consider that such a tree of the BSs is already constructed using any existing approach. In an in-band SNOW tree, nodes are connected to their respective BS through white spaces in a star topology, and BSs are connected, to each other and a root BS, through white spaces in a tree topology. BS_i collects data from its nodes in $SNOW_i$ via a single-hop communication. BS_i forwards the collected information to the root BS (BS_0) via the tree links. In Fig. 3, BS_2 collects information from nodes in $SNOW_2$ and relays it to the root BS via the tree links $BS_2 \rightarrow BS_1 \rightarrow BS_0$. Scalability through in-band integration of multiple SNOWs impacts the *latency*, and hence, we propose to minimize the latency.

Latency along a tree link can be minimized by assigning a large number of subcarriers to it. However, subcarrier assignment on a tree link affects that on other tree links and SNOWs. Also, by transmitting on a large number of subcarriers simultaneously the BS suffers from a traditional OFDM problem called **peak to average power ratio (PAPR)**. PAPR is the ratio of the maximum power of a signal to its average power, and is equal to the number of subcarriers [33], [34]. PAPR can be high after inverse FFT (Fast Fourier Transform) performed at the Tx radio of the BS during downlink transmission due to a large number of subcarriers forcing the non-ideal power amplifier to operate non-linearly. PAPR increases with an increase in the number of simultaneous transmissions on different subcarriers. A

high PAPR impacts the successful reception of a packet and increases power dissipated on the amplifiers resulting in higher power consumption. Therefore, in integration, we have to minimize both latency and PAPR. Note that the (sensor) nodes do not suffer from PAPR as each of them transmits on a single subcarrier.

We minimize both latency and PAPR by assigning an appropriate number of subcarriers to each tree link (for BS-BS communication) and to each SNOW for intra-SNOW communication. This is done at BS_0 that knows the tree topology. We assign subcarriers such that communication along a tree link is not interfered by others. To reduce PAPR, we exclude acknowledgment (ACK) of transmissions along the tree links. SNOW provides a low overhead approach to acknowledge all received packets using one subcarrier. Thus, ACKs can be enabled by adopting SNOW's low-overhead approach of acknowledging all inter-SNOW communications on a single subcarrier. However, we have disabled ACK along inter-SNOW links to keep our approach simple. In our approach, a BS-BS link's reliability can be enhanced by repeating a transmission for critical data flows. In intra-SNOW communication, when a large number of nodes transmit simultaneously to the BS, the BS needs to send ACK to all nodes which may cause high PAPR. To avoid this problem, we combine multiple ACKs into one frame. To handle multiple objectives (latency and PAPR), we will minimize latency while modeling PAPR as a constraint.

3.2 Related Work and New Challenges

To cover a wide area, LoRa integrates multiple gateways through the Internet [35]. Cellular networks do the same relying on wired infrastructure [36], [37]. Rural and remote areas lack such infrastructure. Our proposed in-band integration is conceptually different from and challenging than traditional tiered or clustered wireless sensor network (WSN) [21], [22], [23], wireless sensor networks [24], packet relaying in LPWAN [25], [26] or vehicular networks [27]. First, in these traditional networks, every link operates on its predetermined bandwidth and, hence, bandwidth allocation is not a concern there. In contrast, in our proposed integration, we must assign a bandwidth to each link; these bandwidths among the links can be asymmetric (i.e., of different widths), inter-dependent, and need to be determined optimally or effectively for the sake of network performance. Second, SNOW integration needs to handle the PAPR of BS transmitter, which is absent in those networks. Third, the unique features of SNOW, including massive parallel communication, require a new approach for a seamless integration that will enable concurrent inter- and intra-SNOW communication. Hence, traditional channel allocation for wireless networks [38] or cognitive radio networks [39], [40] cannot be used for bandwidth allocation in SNOW integration.

While SNOW integration was studied in a recent work [28], it does not consider bandwidth allocation among the tree-links of the integration. Namely, every tree-link has the same bandwidth for inter-SNOW communication and it only assigns subcarriers to SNOWs within the integrated network. It tries to maximize the total number of subcarriers across all SNOWs while limiting the number of common

subcarriers between two interfering SNOWs. Thus, it trades scale for inter-SNOW interference. Since all tree links have the same bandwidth, some links closer to the root BS may become congested, affecting the latency and scalability. In contrast, we propose the first scalable in-band integration of multiple SNOWs that minimizes network latency. We minimize the maximum latency through a traffic-aware bandwidth allocation to all links, and by adopting a deterministic MAC protocol that blends with the integration.

3.3 Formulation

The BSs in the proposed integration form a tree topology rooted at BS_0 , where BS_0 has internet access. The parent of BS_i in the SNOW-tree is represented by $BS_{\rho(i)}$, wherein $\rho(i) \in \{0, 1, \dots, N-1\}$. The set of ancestors of BS_i in the SNOW-tree is represented by $BS_k, k \in A(i)$, wherein $A(i) \subseteq \{0, 1, \dots, N-1\}$. Similarly, the set of descendants of BS_i in the SNOW-tree is represented by $BS_k, k \in D(i)$, wherein $D(i) \subseteq \{0, 1, \dots, N-1\}$. We consider that nodes in a SNOW periodically generate a packet. The set of nodes in $SNOW_i$ is represented by V_i , wherein $n_i = |V_i|$ represents the number of nodes in $SNOW_i$. Each node $u \in V_i$ generates packets periodically with a period $T_i(u)$.

The objective of the problem is to minimize the maximum latency to deliver a packet to the root base station (BS_0) among all nodes across all SNOWs, i.e., the integrated network. We want to find the subcarrier assignment for all links that results in the minimum latency.

We consider a uniform subcarrier bandwidth and the set of subcarriers to be orthogonal, enabling parallel communication on the subcarriers. The set of subcarriers is represented by Z . Subcarrier availability at a SNOW depends on the spectrum usage by the primary users, i.e., TV channels and radio users. The set of orthogonal subcarriers available at BS_i is represented by Z_i . The proposed approach assigns a subset of the available subcarriers for inter- and intra-SNOW communication. Specifically, a subset of available subcarriers, $S_i \subset Z_i$, is assigned to BS_i for collecting information from all nodes V_i in $SNOW_i$. $S_{i,j}$ denotes the set of subcarriers allocated for communication along tree link $BS_i \rightarrow BS_{\rho(i)}$, where $BS_{\rho(i)}$ is the parent of BS_i . The set of subcarriers allocated on a tree link must be available at both the parent and child BS, i.e., $S_{i,\rho(i)} \subset Z_i$ and $S_{i,\rho(i)} \subset Z_{\rho(i)}$.

SNOW physical layer uses D-OFDM to enable simultaneous uplink and downlink communication between numerous distributed nodes and the BS. However, the proposed integration becomes *seamless* when inter-SNOW and intra-SNOW communication can happen in parallel. To achieve seamless integration, subcarriers allocated on the tree link should not overlap with those for intra-SNOW communication, i.e., $S_{i,\rho(i)} \cap S_i = S_{i,\rho(i)} \cap S_{\rho(i)} = \emptyset$.

A packet of node u in $SNOW_i$ experiences delay along the link $u \rightarrow BS_i$ and along each tree link from BS_i to BS_0 . For a packet of u of $SNOW_i$, let $\lambda(u, u, BS_i)$ be its estimated intra-SNOW latency under the MAC protocol used in $SNOW_i$ (i.e., the latency to collect the packet at BS_i), and $\lambda(u, BS_j, BS_{\rho(j)})$ be its estimated latency along tree link $BS_j \rightarrow BS_{\rho(j)}$. Then, its total estimated latency $\Lambda(u)$ is given

by Equation (1).

$$\Lambda(u) = \lambda(u, u, BS_i) + \sum_{j \in A(i) - \{0\}} \lambda(u, BS_j, BS_{\rho(j)}) \quad (1)$$

The maximum latency of a packet from SNOW_{*i*} is represented as L_i and is given by the following equation.

$$L(i) = \max\{\Lambda(u) | \forall u \in V_i\} \quad (2)$$

$L(i)$ is an estimate of maximum latency for collecting a packet at root base station BS₀ from SNOW_{*i*}. For fairness, it is important to minimize $\max\{L(i) | 0 \leq i < N\}$ so that data collection from a SNOW does not take overly long time in an integrated network. It also implies minimizing data collection time (e.g., in convergecast scenario) in the integrated network. Therefore, we want to minimize the maximum latency.

The subcarrier allocation must account for interference caused by one SNOW on another SNOW to minimize latency. If some communication in SNOW_{*i*} is interfered by another in SNOW_{*j*}, then SNOW_{*j*} is its **interferer**. Let $I_i \subset \{0, 1, \dots, N-1\}$ be such that each SNOW_{*j*}, $j \in I_i$, is an interferer of SNOW_{*i*}. Let $J_i \subset \{0, 1, \dots, N-1\}$ be such that a transmission along link BS_{*i*} → BS _{$\rho(i)$} can be interfered by that along BS_{*k*} → BS _{$\rho(k)$} or by some node's transmission in SNOW_{*k*}, $k \in J_i$. Constraints $S_{i,\rho(i)} \cap S_j = \emptyset, \forall j \in I_{\rho(i)} \cup \{i, \rho(i)\}$ and $S_{i,\rho(i)} \cap S_{j,\rho(j)} = \emptyset, \forall j \in J_i \cup \{\rho(i)\}$ ensure that tree links are not interfered by any other transmission. We set constraint $\sum_{j \in I_i} |S_i \cap S_j| \leq \sigma_i |S_i|$ to limit the spectrum overlap between two interfering SNOWs where $0 \leq \sigma_i \leq 1$.

Simultaneously transmitting packets on a large number of subcarriers is known to cause PAPR problems, affecting the successful reception of a packet. We overcome the PAPR issue by limiting the number of simultaneous transmissions by a BS. Specifically, we limit the number of subcarriers allocated for transmission to a BS, which includes subcarriers allocated for acknowledging packets and subcarriers allocated along a tree link. We limit the number of subcarriers maximum number of subcarriers allocated to BS_{*i*} to β_i , where β_i is the number of simultaneous transmissions supported by the physical layer of BS_{*i*} with a tolerable PAPR. The value of β_i depends on the saturation point of the BS's transmission amplifier. Considering the BS acknowledges multiple transmissions on a single channel, $\beta_i - 1$ limits the number of concurrent transmissions by a BS_{*i*} on the tree link. Thus, we use constraint $1 \leq |S_{i,\rho(i)}| < \beta_i$. Our objective is to determine S_i and $S_{i,\rho(i)}$ for all i so as to

$$\text{Minimize } \max\{L(i) | 0 \leq i < N\}$$

$$\text{subject to } S_i \subseteq Z_i, S_{i,\rho(i)} \subseteq Z_i, S_{i,\rho(i)} \subseteq Z_{\rho(i)} \quad (3)$$

$$S_{i,\rho(i)} \cap S_j = \emptyset, \forall j \in I_{\rho(i)} \cup \{i, \rho(i)\} \quad (4)$$

$$S_{i,\rho(i)} \cap S_{j,\rho(j)} = \emptyset, \forall j \in J_i \cup \{\rho(i)\} \quad (5)$$

$$\sum_{j \in I_i} |S_i \cap S_j| \leq \sigma_i |S_i|; 1 \leq |S_{i,\rho(i)}| < \beta_i \quad (6)$$

Fig. 4 shows an illustration of decision variables for the tree of Fig. 3. Note that the above formulation can be executed whenever there is a change in white spaces' availability. Typically, change in white space availability is less frequent as the incumbents (such as primary users like TV-broadcasters) do not vary often. A channel availability may change once in several days or less frequently. Fur-

thermore, the above formulation is applicable for any MAC protocol used inside each SNOW. However, network latency expressions are usually non-linear and non-differentiable, and so does the above optimization problem [41].

Adopting a sub-gradient approach or some global optimization framework such as Simulated Annealing based penalty method [42] for an optimal solution can be extremely time consuming. Spectrum dynamics of white spaces requires a frequent re-run of the optimization. Instead, we propose a greedy heuristic that quickly provides solution through a traffic-aware and latency-aware spectrum allocation along the links to achieve an integrated network. The heuristic iteratively identifies subcarriers that help reduce the network's latency, and assigns those to the links. The heuristic requires a good latency estimation to identify latency-minimizing subcarriers. Hence, we first describe SNOW MAC protocol and derive its latency expression, and then describe the spectrum allocation algorithm.

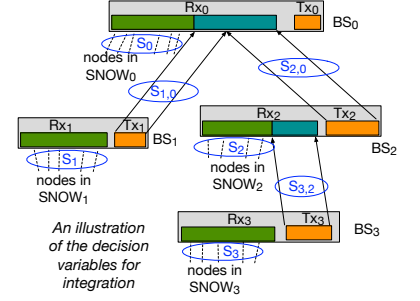


Fig. 4: Integration using dual-radios.

4 DESCRIPTION OF THE LOW-LATENCY INTEGRATION OF MULTIPLE SNOWS

Long-range communications on CSMA/CA and ALOHA MAC protocols are prone to an unprecedented number of packet collisions at the base stations due to simultaneous packet transmission from many nodes that are hidden from each other [43]. Thus, CSMA/CA and ALOHA-based MAC protocols cause unpredictable latencies. These MACs are less preferred for applications requiring low latencies. On the contrary, TDMA (time division multiple access) based MACs offer collision-free communication between SNOW nodes and a BS, ensuring predictable latencies. Thus, we propose to use a time-triggered MAC protocol based on TDMA inside each SNOW. In the following sections, we will describe two MAC protocols, one for exclusively time-triggered traffic, namely TDMA MAC, and another for a combination of time-triggered and event-triggered traffic, namely RI-TDMA (Receiver Initiated TDMA) MAC. We also derive the latency estimations for both MAC protocols.

4.1 Low-Latency TDMA MAC

Here, we describe a simple low-latency TDMA MAC that supports time-triggered traffic within a SNOW integration.

4.1.1 TDMA MAC for SNOW Integration

In TDMA MAC for SNOW Integration, all nodes and BSs are time-synchronized, and time in the network is divided into time slots. A packet transmission and its acknowledgment happen in one time slot. Each node in a SNOW is assigned a time slot and a subcarrier for transmission. Multiple nodes can transmit packets to their BSs on unique subcarriers in the same time slot. The BS acknowledges all transmissions

received in a time slot via a single ACK message. The ACK message consists of a bit vector of size equal to the number of available subcarriers, where the i^{th} bit of the bit vector is used to ACK the transmission on the i^{th} subcarrier. If the BS receives a packet on a subcarrier, the corresponding bit in the bit vector is set to 1. Otherwise, it remains 0. Nodes can use the information about their assigned subcarrier to infer their packet transmission status.

Energy saving and sleep. Nodes are power-constrained and are considered to be sleeping all the time. They wake up for sensing and transmission during their scheduled time slot.

Scheduling packets/nodes. In SNOW, a BS can receive simultaneously on any number of the available subcarriers. Thus, the BS can consider each available subcarrier as a separate processor and adopt existing multiprocessor scheduling algorithms like Global EDF (earliest deadline first) or Global RM (rate monotonic) to generate a transmission schedule. The transmission schedule of the nodes' time slot and subcarrier assignment. Upon generating a schedule, the BS disseminates the schedule to all nodes in its SNOW. Typically, the BS generates and disseminates the transmission schedule during network deployment.

Scheduling Communication between Two BSs. To facilitate a seamless SNOW integration, a BS forwards packets to its parent BS on dedicated subcarriers with no other scheduled packet transmissions. Since a BS has two radios, it can transmit and receive packets simultaneously. A BS in SNOW transmits packets simultaneously on β subcarriers with a tolerable PAPR. Here, the value of β depends on the saturation point of the BS amplifier. SNOW BS typically uses USRP devices for the physical layer. Typically, USRP devices support simultaneous transmission on many subcarriers with a tolerable PAPR. Within a time slot, a BS acknowledges all received packet transmissions on one subcarrier. It may utilize up to $\beta - 1$ common subcarriers with its parent BS to forward packets along the tree link. Specifically, it will forward up to $\beta - 1$ packet received in the previous time slot to its parent BS. The BS selects packets for forwarding based on the EDF or RM scheduling policy. These transmission along the tree link happen on unique channels, and hence, they do not interrupt information collection from SNOW nodes. Since a BS can receive packets concurrently on all the allocated subcarriers, the parent BS can receive all transmissions along the tree link and its nodes concurrently.

Subcarrier sharing between BSs. To improve the scalability of the network, BSs may share a common subcarrier. Sharing a subcarrier between two BSs that are in communication range of each other can cause packet collisions. To minimize collisions, the root BS generates a round robin schedule for subcarrier sharing and disseminates the schedule to all BSs. **Transmission failures and retransmissions.** Although TDMA-based MAC eliminates collisions, poor link qualities can cause packet transmission failures. To improve communication reliability, the BS has an option to schedule multiple time slots for a transmission from each node. A node can use the additional time slots for retransmissions.

4.1.2 Latency Estimation for TDMA MAC

Here, we provide an approach for estimating packet latencies. We estimate the worst-case latency for a special case

where all the nodes use the same periods, one-time slot for transmission, and no subcarrier sharing between BSs. We choose to estimate for the special case since the latency estimation is easy to analyze, resulting in a fast estimation and reducing the computational overhead of subcarrier allocation on the root BS.

In TDMA MAC, a packet from node u experiences the maximum delay when packets from all nodes in SNOW $_i$ interfere it. Considering n_i is the number of nodes in SNOW $_i$ and $|S_i|$ is the number of subcarriers allocated to SNOW $_i$, the maximum latency experienced by a packet from node u to reach BS $_i$ is given as follows.

$$\left\lceil \frac{n_i}{|S_i|} \right\rceil \quad (7)$$

A packet generated by u experiences the maximum delay along the tree link BS $_j \rightarrow$ BS $_{\rho(j)}$ when packets from all nodes that pass through the BS $_j$ interfere it. Considering $|S_{j,\rho(j)}|$ is the number of common subcarriers between BS $_j$ and BS $_{\rho(j)}$ and $D(j)$ is the set of SNOWs that are descendants of SNOW $_j$, the maximum latency experienced by a packet generated by u along the tree link BS $_j \rightarrow$ BS $_{\rho(j)}$ is given by the following equation:

$$\left\lceil \frac{\sum_{k \in D(j)} n_k}{|S_{j,\rho(j)}|} \right\rceil. \quad (8)$$

Adding the latency experienced at each BS along the tree, the maximum latency incurred to transmit from a node in SNOW $_i$ to the root BS is given by Equation (9).

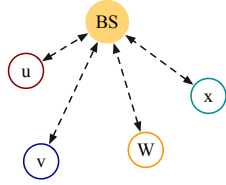
$$L(i) = \begin{cases} \left\lceil \frac{n_i}{|S_i|} \right\rceil, & \text{if } i = 0; \\ \left\lceil \frac{n_i}{|S_i|} \right\rceil + \sum_{j \in A(i) - \{0\}} \left\lceil \frac{\sum_{k \in D(j)} n_k}{|S_{j,\rho(j)}|} \right\rceil, & \text{otherwise.} \end{cases} \quad (9)$$

The above latency formulation is a special case of the maximum latency derived in the previous section, without an efficient method of subcarrier sharing and assuming all nodes have the same periods. Although these assumptions restrict the applicability, it provides a tighter bound on the maximum latency computation and provides a better solution to the subcarrier allocation problem.

4.2 RI-TDMA MAC for SNOW Integration

Although the low-latency TDMA MAC proposed in the previous section achieves predictable latencies, it has the following limitations. It can only support time-triggered traffic but not event-triggered traffic. TDMA-based protocols require strong time synchronization, which consumes high energy at the nodes. To handle packet failures, the central manager allocates multiple time slots for each packet. Typically, schedules with pre-determined redundancy causes wastage of time slots and impacts the scalability of the system [44]. Thus, TDMA MAC is only applicable for special applications.

To overcome the above limitations, we also propose a receiver-initiated MAC for TDMA, RI-TDMA for general applications. In RI-TDMA MAC, the BS schedules nodes for transmission in each slot via a request message. A receiver-initiated message eliminates the need for executing a time synchronization protocol at the node. Furthermore, it can support event-triggered traffic and handles transmission



(a) An Example of a Single SNOW.

	1		2		3		Time Slot
BS	Req-{u,v}	Listen	Req-{w,x}	Listen			
u	Listen	Tx	Listen				Sleep
v	Listen	Tx	Listen				Sleep
w			Sleep	Listen	Tx	Listen	Sleep
x			Sleep	Listen	Tx	Listen	Sleep

(b) Transmission Schedule for the Example Network.

Fig. 5: An Example of Packet Scheduling in RI-TDMA MAC.

failures and packet re-transmissions dynamically, thereby making it applicable for a wide range of applications.

RI-TDMA is a TDMA adaptation of the prominent receiver-initiated MAC protocols. Under RI-TDMA, the SNOW BS operates in time slots. Each time slot contains two stages: a request stage and a data transmission stage. During the request stage, the BS schedules a subset of nodes for transmission on the available subcarriers by transmitting a request message. During the data transmission stage, the scheduled nodes respond by transmitting data. Although RI-TDMA operates in time slots, only the BSs have to be time synchronized. We consider that BSs are not energy-constrained, and hence their energy consumption is out of our consideration. Nodes need not be time synchronized as they only respond with data upon receiving a request message. Note that the length of a time slot can be calculated as the sum of the request message's transmission time, data transmission time from a node, twice the subcarrier switching time of the node, and a buffer time of 2-3ms.

An example of data transmission in RI-TDMA MAC between a BS and four nodes is shown in Fig 5. Nodes u, v, w, x have packets to transmit at time 0, and the BS has two subcarriers on which it can receive simultaneously. During the request stage of time slot 1, the BS requests data from u and v . During the data transmission stage of time slot 1, u and v respond with the data. Similarly, during time slot 2, the BS requests data from w and x , and within the same time slot, w and x respond with the data. Since the nodes are not time synchronized, the nodes may start listening or transmitting at a different time, as shown in Fig. 5b.

Subcarrier sharing between BSs. To improve the scalability of the network, BSs can share a common subcarrier. Sharing a subcarrier between two BSs that are in communication range of each other can cause packet collisions. To minimize such collisions, they use round-robin to share a subcarrier such that each BS gets uninterrupted access to the subcarrier for an equal percentage of time slots. For example, if two BSs (BS_0 and BS_1) are sharing one subcarrier, then BS_0 will use the subcarrier for time slots 1, 3, 5, 7, \dots and BS_1 will use the subcarrier for time slots 2, 4, 6, 8, \dots . To determine the time slots with uninterrupted access to a common subcarrier, BSs using the same subcarrier first form an interference graph, where two BSs in the communication range of each other have an edge in the interference graph. BSs can use vertex coloring on the interference graph. Each color represents

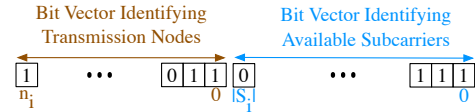


Fig. 6: Payload from BS_i 's Request Message.

a time slot in a periodically repeating subcarrier sharing schedule. The subcarrier sharing schedule repeats every epoch and the length of the subcarrier sharing epoch is the same as the maximum number of colors which is upper bounded by the maximum degree in the interference graph. We choose the length of the subcarrier sharing epoch to be harmonic to the transmission period of the nodes. Since the BSs in the network are time-synchronized, each node knows the start of a subcarrier sharing epoch and the time slot during which it has uninterrupted access to the subcarrier. Note that the algorithm to compute the subcarrier sharing schedule can be executed distributedly at BSs or centrally at the root BS.

Scheduling packets/nodes within a time slot. The BS of a SNOW schedules the transmissions in a time slot using the rate monotonic (RM) policy. In a time slot, the BS prioritizes the packets that have not been received based on their periods, the packet (and its associated node) with the shortest period being assigned the highest priority. Any tie is broken using node ID, the node with the smaller ID being assigned a higher priority. The BS then schedules the first α packets for transmission in that slot, where α is the number of subcarriers available in that time slot. Note that such scheduling is dynamic and happens at the beginning of the time slot.

Scheduling nodes through request message. Upon identifying the transmitting nodes in a time slot, a BS requests information from all transmitting nodes on a single subcarrier called the *downlink subcarrier*. To identify the transmitting nodes, the BS uses a bit vector of size equal to the number of nodes and sets the bit to 1 for the transmitting nodes. To identify a set of available subcarriers in a time slot, it uses another bit vector of size equal to the number of subcarriers and sets the bit of corresponding available subcarriers to 1, as shown in Fig. 6. To identify the subcarrier on which a node transmits a packet, it uses the following relation: the α -th transmitting node from bit vector (identifying transmitting nodes) uses the α -th available subcarrier indicated in the subcarrier bit vector. For example, if a BS sets bits 2-6 of the bit vector identifying transmitting nodes to 1 and 0-3 bits and 5-th bit of the subcarrier bit vector are set to 1, then the 4-th transmitting node, i.e., node 6 sends the packet on the 4-th available subcarrier, i.e., subcarrier 5. Note that the mapping between the subcarrier ID and the actual frequency is communicated during network deployment, similar to SNOW MAC in [28].

Transmission failures and retransmissions. Although RI-TDMA MAC minimizes packet collisions, the BS may not receive some packets due to bad link qualities. To increase the reliability in communication, we use request message of the next slot as an ACK. If the BS requests a packet from a node in the next slot (negative-ACK), then the node assumes that the BS did not receive the packet and re-transmits it. If the node receives a receiver-initiation message without a request for transmission directed to it (positive-ACK), it assumes the packet transmission was successful, as shown

in Fig. 5b. If the node does not receive a receiver-initiation message (no-ACK), then it listens to subsequent time slots until it receives a positive-ACK or a negative-ACK. Since the BS dynamically schedules packets at each time slot, a failed transmission can be re-scheduled for transmission on the next time slot. Furthermore, a failed packet can be retransmitted multiple times if needed. Note that the BS chooses to re-transmit a higher priority packet before the transmission of a lower priority packet. Due to the dynamic nature of scheduling and packet re-transmission, RI-TDMA MAC is scalable with the number of nodes. Note that the receiver initiation message serves the purpose of an ACK, and hence, it does not pose any additional impact on the PAPR of the BS.

Handling persistent transmission failures on a subcarrier. When a BS detects persistent transmission failures on a subcarrier, it can blacklist that subcarrier. That is, a BS does not request for transmissions on blacklisted subcarriers. Note that a persistent transmission failure can be caused by temporary events like obstacles or noise. To resume operations on a blacklisted subcarrier, after several hours of being blacklisted, BS can temporarily request transmissions on the blacklisted subcarrier. If multiple transmissions are successful, the subcarrier is removed from the list of blacklisted subcarriers. If multiple transmissions fail, the subcarrier is blacklisted again. Note that the blacklisting of a subcarrier is local to a SNOW, and hence, does not require global synchronization.

Energy saving and sleep. To conserve energy, nodes in RI-TDMA MAC wake up for packet generation/transmission and sleep for the rest. Since RM generates a compact schedule consisting of a transmission time slot for the first packet, nodes can use the first transmission time slot without any retransmissions and period to predict the next transmission time slot. Note that the next wake up may not be precise due to (1) packet retransmissions delaying the schedule, and (2) clock drifts. Due to unpredictable nature of packet transmissions, a node cannot precisely estimate its next transmission time slot and wait until the BS sends a request message. However, a node can compensate for the clock drift locally by measuring the error between the start of the receiver-initiated message and its wake-up time. The goal of the clock drift estimation is to enable nodes to wake up closer to the start of the receiver-initiated message but not for clock synchronization. Note that the node may have to wait for multiple receiver-initiated messages before it receives a request message from the BS. Furthermore, the clock drift estimation provides a weak time synchronization. However, the use of receiver-initiation eliminates the need for executing a time synchronization protocol to strongly synchronize the nodes.

4.2.1 Scheduling Event-Triggered Nodes

Typically, a node periodically senses the state of an environment and transmits the sensed information to its BS. However, some sensor nodes may generate packets on the occurrence of a specific event. We refer to these nodes as *event-triggered nodes*. In this paper, we consider that most nodes use time-triggered sensing, and some nodes use event-triggered sensing.

Typically, a BS cannot predict the occurrences of events, and hence, cannot schedule an event-triggered node for a packet transmission. To address this challenge, we leverage on the concurrent transmission and reception capabilities of the BS. Specifically, during the request stage of a time slot, the BS is not listening for packet transmissions, and hence, many subcarriers (except for the downlink subcarrier) are unused. Event-triggered nodes transmit packets to the BS on any such unused subcarrier during the request stage.

Since nodes are not time-synchronized with the BS, event-triggered nodes cannot predict the start of a request stage of a time slot. To overcome this challenge, an event-triggered node first listens to one packet transmission request from the BS and uses the length of the time slot to predict the start of the next request stage. Thus, an event-triggered node can communicate with the BS without interfering with the network's time-triggered transmission. Furthermore, the BS can acknowledge the packet transmission on its downlink subcarrier during the data transmission segment while time-triggered nodes transmit packets to it.

Typically, a single SNOW BS may communicate with multiple event-triggered nodes. However, the probability of all event-triggered nodes transmitting a packet at the same time is very low. If they transmit at the same time and collide, the nodes use a fixed random back-off interval for a re-transmission.

4.2.2 Scheduling Communication between Two BSs

To facilitate a seamless integration, a BS forwards packets to its parent BS on dedicated subcarriers that are not interfered by any other communication in the network. Since a BS is equipped with two radios, it can transmit and receive packets simultaneously. Since the inter-snow communication happens on a dedicated subcarrier, BSs do not use receiver-initiated communication on the tree links. Instead, the BS forwards the packets to its parent on the next available subcarrier and time slot.

In RI-TDMA MAC, a BS uses at most one subcarrier in both request and data transmission stages. Thus, it can transmit packets on the tree link subcarriers during both stages of the time slot. Specifically, a BS may transmit up to $\beta - 1$ packets during each stage of the time slot with an acceptable PAPR.

A BS locally determines the following packets for forwarding based on the packets' priority. Unlike SNOW nodes, the BS has sufficient energy and memory, and hence, it may execute scheduling algorithms locally to determine packets for forwarding locally.

Similar to intra-SNOW communication, packets in inter-SNOW communication are prioritized based on the RM scheduling policy. A packet generated by an event-triggered node typically has a lower priority than a packet generated by a time-triggered node. Thus, the BS prioritizes packets from time-triggered nodes over packets from event-triggered nodes.

4.2.3 Latency Estimation for RI-TDMA

Here, we provide an approach to estimate packet latencies. The latency estimate for RI-TDMA can be used for a latency-aware subcarrier allocation for inter- and intra-SNOW communication to achieve a low-latency integration

of multiple SNOWs. Note that it is highly challenging in a wireless environment to guarantee a packet's successful reception and bound the maximum latency on a link. Hence, for an effective subcarrier allocation, we instead rely on a good latency estimate that represents latency of most common cases, instead of a worst-case upper bound. Specifically, we determine a latency bound without considering re-transmission. Under no re-transmission, the latency expression provides a good estimate since (1) the adopted RI-TDMA MAC avoids collisions for time-triggered packets and minimizes collisions for event-triggered packets, and (2) subcarriers with poor link quality can be blacklisted during network deployment to maximize the number of successful transmissions in the first attempt.

The first step in estimating the maximum latency of packet from u of SNOW_i is to determine the intra-SNOW latency from u to BS_i , $\lambda(u, u, \text{BS}_i)$. To determine the intra-SNOW latency, we rely on the concept of critical instant, similar to processor scheduling. The critical instant, in RI-TDMA, is the arrival of a packet such that it experiences the maximum delay from all higher priority packets. Note that the high priority packets of v denoted by $hp(v)$ are those generated by a node v with periods smaller than or equal to the period of u , i.e., $v \in hp_i(u)$ if $v \in V_i$ and $T_i(v) < T_i(u)$, where $T_i(v)$ represents the period of node v , $v \in \text{SNOW}_i$. Since BSs use RM scheduler, the maximum delay occurs when all high priority packets arrive at the same time as the packet from u . During a critical instant, multiple packets of a high priority node can delay one packet of u . The maximum number of packets generated by a high priority node with an interval of length $\lambda(u, u, \text{BS}_i)$ is given by the following expression:

$$\sum_{v \in hp_i(u)} \left\lceil \frac{\lambda(u, u, \text{BS}_i)}{T_i(v)} \right\rceil. \quad (10)$$

Each packet of a high priority node delays the packet of node u by one time slot.

In RI-TDMA, BS_i can receive from at most $|S_i|$ nodes concurrently, where S_i represents the set of subcarriers assigned to BS_i . However, BS_i may share a subset of S_i subcarriers with other BSs, and sharing allows BS_i to transmit packet on these shared subcarriers for a fraction of the time. We represent the average number of subcarriers available to BS_i within a time slot as ψ_i . Since BS_i can receive ψ_i packets within one time slot, the number of time slots required to transmit L packets is $\lceil \frac{L}{\psi_i} \rceil$. Thus, the number of time slots required to transmit all high priority packets is shown below

$$\left\lceil \frac{1}{\psi_i} \sum_{v \in hp_i(u)} \left\lceil \frac{\lambda(u, u, \text{BS}_i)}{T_i(v)} \right\rceil \right\rceil \quad (11)$$

Note that the concept of subcarrier sharing between BSs adds an additional latency. For example, if one subcarrier is available at BS_i , it can receive a packet every time slot. However, if 4 subcarriers each with a fractional availability of $\frac{1}{4}$ are available to BS_i , the average number of subcarriers available to BS_i is 1, but the BS_i can receive 4 packets simultaneously at the 4th slot as shown in Fig 7. As shown in this example, subcarrier sharing can cause an additional delay on a packet. This additional delay, represented by ς , is the maximum period of subcarrier sharing (which is 4 in the above example).

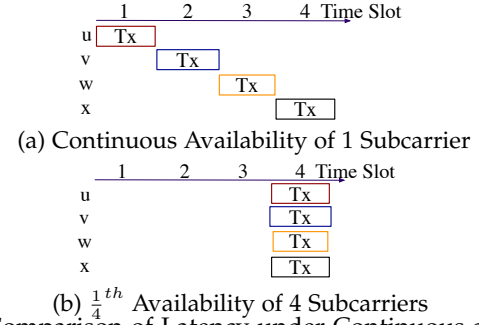


Fig. 7: Comparison of Latency under Continuous and Fractional Availability of a Subcarrier.

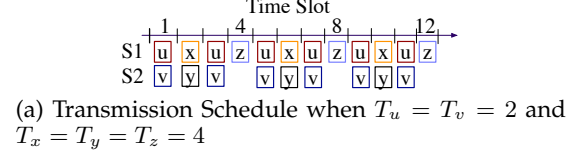


Fig. 8: Example of Schedule when Period of One Node increases by 1 Time Slot.

The maximum latency experienced by a packet of u is the sum of the number of transmission attempts from u (which is one) and the total delay caused by all high priority packets. The solution to the expression in Equation (12) gives the maximum latency experienced by a packet from node u .

$$\lambda(u, u, \text{BS}_i) = 1 + \varsigma + \left\lceil \frac{1}{\psi_i} \sum_{v \in hp_i(u)} \left\lceil \frac{\lambda(u, u, \text{BS}_i)}{T_i(v)} \right\rceil \right\rceil \quad (12)$$

Note that scheduling packets for intra-SNOW communication in RI-TDMA is a special case of a task scheduling on a multi-core processor, where each packet transmission (or equivalent task) executes for precisely one time slot. Here, increasing the period of a node by one time slot, facilitates a lower priority packet from to start and complete transmission within this time slot. The transmission of the lower priority packet earlier causes a ripple effect on other lower priority tasks to finish earlier or at the same time as before. For example, consider a BS_1 has two available subcarriers and has to schedule packets from 5 nodes, u , v , x , y , and z . Packet transmission schedule when $T_u = T_v = 2$ and $T_x = T_y = T_z = 4$ is shown in Fig. 8a. When the period of u is changed to 3 (i.e., $T_u = 3$), BS_1 can schedule a packet z on time slot 2, as shown in Fig 8b. The scheduling of z 's packet earlier causes a ripple effect freeing 2 time slots in the hyperperiod of 12 time slots, as shown in Fig. 8b. Thus, increasing the period of a node does not increase the latency of any lower priority packet, and scheduling anomalies discussed in [45] are not applicable here. Similarly, packet scheduling on a fractional availability of a subcarrier, does not introduce any scheduling anomalies for packet scheduling in RI-TDMA MAC. Thus, the expression in Equation (12) provides the maximum latency experienced by u 's packet.

The next step in estimating the latency of a packet is to compute the inter-SNOW latency. We first describe the latency experienced by a packet (originating at node u) to reach $\text{BS}_{\rho(i)}$ from BS_i , where node u is in SNOW_i (i.e., $u \in$

V_i). We then extend the description for any BSs BS_j and $BS_{\rho(j)}$, along the tree path of BS_i .

In inter-SNOW communication, a packet experiences delays from two sources: (1) high priority packets that previously delayed this packet, and (2) new high priority packets that arrive at BS_i . The first source of delay emanates from the mismatch in the subcarrier assignment for intra-SNOW communication ($|S_i|$) and inter-SNOW communication ($|S_{i,\rho(i)}|$). Specifically, $|S_i|$ can be higher than $|S_{i,\rho(i)}|$, i.e., BS_i can receive packets at a higher rate than it can transmit to its parent. Thus, causing the high priority packets to further delay the packet from u . The latency caused by high priority packets that previously delayed a packet of u is given by the following expression:

$$\lambda(u, u, BS_i) \left[\frac{\psi_i}{2\epsilon|S_{i,\rho(i)}|} \right] \quad (13)$$

Here, $2\epsilon|S_{i,\rho(i)}|$ represents the number of concurrent transmissions BS_i can make in a time slot. The 2ϵ transmissions arises (1) due to the disparity in the subcarrier bandwidth allocated to the nodes and tree links, and (2) since the BS can transmit on both stages of the time slot.

The second source of delay emanates from the arrival of new high priority packets at BS_i . The number of high priority packets arriving at BS_i depends on the generation rate of the high priority packets and the packet reception rate of these packets at BS_i . For example, if BS_k receives 10 high priority packets but can transmit only 6 in one time slot to BS_i , then a packet from u can be interfered by 6 new high priority packets from BS_k in one time slot. Let $C(i) \subseteq \{0, 1, \dots, N-1\}$ be such that each $B_k, k \in C(i)$, is a child of BS_i in the SNOW tree. An upper bound on the number of new high priority packets arriving at BS_i during the interval $\lambda(u, BS_i, BS_{\rho(i)})$ is given by the following Equation:

$$\left[\sum_{\substack{v \in hp_g(u) \ \& \ g \in D(k) \\ k \in C(i)}} \frac{\lambda(u, BS_i, BS_{\rho(i)})}{T_g(v) |S_{k,i}|} + \sum_{w \in hp_i(u)} \frac{\lambda(u, BS_i, BS_{\rho(i)})}{T_i(w) \psi_i} \right] \quad (14)$$

In the above expression, the first summation takes into account all high priority packets that are generated by a successor of BS_i and second summation takes into account all the high priority packets generated by nodes of $SNOW_i$.

If the total number of subcarriers on which BS_i receives a packet in one time slot is less than $2\epsilon|S_{i,\rho(i)}|$, then it cannot transmit $2\epsilon|S_{i,\rho(i)}|$ packets in a time slot. Thus, we represent the maximum number of concurrent inter-SNOW transmissions by BS_i as $\phi_{i,\rho(i)}$, and is given by Equation (15).

$$\phi_{i,\rho(i)} = \min(2\epsilon|S_{i,\rho(i)}|, \psi_i + \sum_{k \in C(i)} |S_{k,i}|) \quad (15)$$

Thus, the maximum latency experienced by a packet of u to reach $BS_{\rho(i)}$ from BS_i is given by the sum of two sources. The maximum latency experienced at BS_i by a packet from u is given in Equation (16).

$$\lambda(u, BS_i, BS_{\rho(i)}) = \lambda(u, u, BS_i) \left[\frac{\psi_i}{2\epsilon|S_{i,\rho(i)}|} \right] + \left[\frac{1}{\phi_{i,\rho(i)}} \left[\sum_{\substack{v \in hp_g(u) \ \& \ g \in D(k) \\ k \in C(i)}} \frac{\lambda(u, BS_i, BS_{\rho(i)})}{T_g(v) |S_{k,i}|} + \sum_{w \in hp_i(u)} \frac{\lambda(u, BS_i, BS_{\rho(i)})}{T_i(w) \psi_i} \right] \right] \quad (16)$$

Algorithm 1: LT-SASI

Input : Tree Structure, $Z_i, I_i, J_i, \forall i \in \{0, N\}$
 $T_u \forall u \in BS_i, 0 \leq i < N$

Output: $S = \cup_{\forall i} \{S_i, S_{i,\rho(i)}\}$

```

1 for  $i \in \{0, N\}$  do
2    $S_i = \text{Unique\_Subcarrier}(Z_i, I_i, S)$ 
3    $S_{i,\rho(i)} = \text{Unique\_Subcarrier}(Z_i, Z_{\rho(i)}, J_i, S)$ 
4 end
5 do
6    $L = \text{Compute\_Latency}()$ 
7    $A = \text{Exclude\_infeasible\_links}(L, S)$ 
8   if  $|A| = 0$  then
9     break
10  end
11   $v, \nu = \text{Identify\_Bottleneck\_Link}(A)$ 
12  if  $v \in BS$  then
13     $S_{v,\nu} = \text{Add\_Feasible\_Channel}(S, Z)$ 
14  else
15     $S_\nu = \text{Add\_Feasible\_Channel}(S, Z)$ 
16  end
17 while True

```

The maximum latency experienced by a packet from u at some BS_j can be computed similar to Equation (16) by replacing the transmission from node BS_i to BS_j with the transmission from BS_j to $BS_{\rho(j)}$.

The maximum latency experienced by a packet generated by node u to reach the root BS is the summation of latency experienced to reach u 's BS BS_i and all links from on the tree from BS_i to root BS. It is given by Equation (1), which is:

$$\Lambda(u) = \lambda(u, u, BS_i) + \sum_{j \in A(i) - \{0\}} \lambda(u, BS_j, BS_{\rho(j)}). \quad (17)$$

Latency estimation for event-triggered packets. Assuming an event triggered packet has the lowest priority when compared to all time-triggered packets, the latency of event triggered packets can be computed from Equation (1).

The aforementioned latency formulation can be substituted in the problem formulation described in Section 3.3. Since the latency computation takes a pseudo-polynomial time to generate an accurate result, generating an optimal solution using any existing optimization solvers like genetic algorithm takes a significantly long time. Hence, we propose a heuristic solution to generate an efficient solution.

4.3 Latency and Traffic aware Spectrum Allocation for SNOW Integration (LT-SASI)

The heuristic iteratively identifies subcarriers that help reduce the network's latency, and assigns those to the links. In the greedy heuristic, we start by allocating one subcarrier to each link in the network. We then compute the bottleneck link (a link that causes the largest delay on the packet with maximum latency). We assign one subcarrier to the bottleneck link and repeat the process until no further allocation is possible. We refer to this greedy heuristic as LT-SASI and present as Algorithm 1.

The heuristic starts by assigning one unique subcarrier to each inter-SNOW link and intra-SNOW link, as shown in Lines 1-4. The function `Unique_Subcarrier` first computes the list of subcarriers not used by any interfering SNOW and returns one subcarrier from the unused subcarrier list. The heuristic then computes all packets' maximum

latency in the network, as shown in Line 6. The function `Compute_Latency` uses a polynomial-time simplification of Equation (1) by replacing the critical window length with the period of the packet to compute the latency of all nodes in the network. Aspirant links are then calculated by eliminating links that do not have any feasible subcarrier assignment that minimizes latency, as shown in Line 7.

The function `Exclude_infeasible_links` first computes the list of available subcarriers for each link. It then excludes any link that has no feasible subcarrier. If there are no feasible subcarriers for any link, the greedy heuristic stops executing. Otherwise, it finds the bottleneck link, as shown in Line 11. The function `Identify_Bottleneck_Link` finds the node with the highest latency and the link along the route with the highest latency. For the bottleneck link, the greedy heuristic then finds a feasible subcarrier for the bottleneck link using the function `Add_Feasible_Channel`. The function `Add_Feasible_Channel` finds an available subcarrier that minimizes latency and meets the Constraints (3), (4), (5), and (6). The greedy heuristic then assigns the feasible subcarrier to the bottleneck link, as shown in Lines 12-16.

The root BS executes the LT-SASI algorithm and disseminates the subcarrier assignment to all BSs. Upon a change in the available white space spectrum or packet transmission periods, the root BS re-computes and re-disseminates the subcarrier assignment. The root-BS assigns stable subcarriers (that do not change frequently) for inter-SNOW communication to facilitate the dissemination of new subcarrier assignment. The root-BS can use any existing dissemination protocol. For simplicity, we use the TinyOS dissemination protocol. Since we use a tree, each BS relays information only to its child BSs. Thus, the latency overhead for dissemination is in the order of the depth of the SNOW tree. Note that the re-dissemination of subcarrier information obviates the need for a global agreement.

Theorem 1. *The time complexity of LT-SASI algorithm, presented in Algorithm 1, is $O(N^2|V||Z| + N^2|Z|^2)$.*

Proof. The `Unique_Subcarrier` function selects one subcarrier from a list of subcarriers Z , and hence, its time complexity is $O(|Z|)$. The initial subcarrier assignment iterates over `Unique_Subcarrier` function for each of the N SNOW BSs, and hence, the time complexity for the initial subcarrier assignment is $O(N|Z|)$. The subsequent additions of the channels requires $N|Z|$ iterations over the functions `Add_Feasible_channel`, `Exclude_infeasible_links`, `Identify_Bottleneck_link`, and `Compute_Latency`, and hence, it is the primary source of time complexity. The time complexity of computing the maximum latency for all nodes is $O(N|V|)$ since `Compute_Latency` function iterates over N SNOWs and $|V|$ nodes in a SNOW as shown in Equation (12). The time complexity of `Exclude_infeasible_links` function is $O(N|Z|)$, considering that the function iteratively tries to find a feasible subcarrier for each link. The time complexity of `Identify_Bottleneck_link` is $O(N)$ since the maximum depth of the tree is N . The time complexity of `Add_Feasible_channel` that minimizes the latency of the bottleneck link is $O(|Z|)$. The time complexity of one

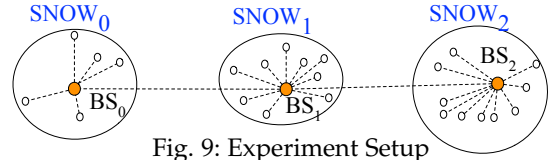


Fig. 9: Experiment Setup

iteration of adding one subsequent channel simplifies to $O(N|V| + N|Z|)$. Thus, the time complexity of LT-SASI algorithm is $O(N^2|V||Z| + N^2|Z|^2)$. \square

5 EXPERIMENT

Here, we evaluate the proposed low-latency integration of SNOWs with RI-TDMA MAC through real experiments. We consider application scenarios such as monitoring oil fields and agricultural IoT for the evaluation. We compare the performance of the proposed approach with the greedy heuristic for scalability optimization (SOP) proposed in [28], referred to as *Greedy-SOP* in this paper. Note that, in this paper, we adopted new MAC protocols, RI-TDMA and Low-Latency TDMA, that blends with the low-latency integration. Greedy SOP integration approach is designed for CSMA/CA based MAC as it trades scale for inter-SNOW interference experienced in the CSMA/CA protocol. Adopting a different MAC protocol would require a significant modification to this baseline integration approach. Therefore, to retain the originality of the baseline approach, we cannot compare the two integration approaches considering the same MAC protocol. TDMA MAC needs strong time synchronization between the SNOW nodes that is not developed for the CC1310 radio. Thus, we evaluated only RI-TDMA MAC in the experiments. A comprehensive evaluation with TDMA MAC is presented in the simulation.

We create a SNOW integration with 3 BSs, as shown in Fig. 9. We performed large scale evaluation with a wide range of scenarios considering random tree structures in the simulations (Section 6). For the experiment, BS_0 and BS_2 are placed 1km away from BS_1 such that BS_0 and BS_2 communicate with BS_1 but not with each other. In the experiment, each BS uses 2 USRP devices –one for Tx radio and the other for Rx radio. We have adopted the open-source implementation of SNOW that is available at [16]. Each SNOW network is assigned 5 CC1310 devices [46]. Each CC1310 node transmits packets to their SNOW BS, and these transmissions are susceptible to packet drop due to noise. Each of the 15 CC1310 devices emulates the communication pattern of 4-20 devices (depending on the setup). After a successful packet transmission, each device hops to the next subcarrier and transmits a packet at the next available time. Note that, in this setup, all interfering nodes operate on the same channel at the same time, taking into account the impact of packet collisions. After successfully transmitting a packet, they hop to a different channel and interfere with each other again. The emulation of nodes shows the working of LT-SASI under a large-scale integration. Note that a corresponding large-scale experiment with an actual number of devices will exhibit (almost) similar performance.

We used 28 subcarriers between 500 MHz and 512 MHz from 2 TV channels of the white space spectrum. Both CC1310 and USRP devices use a transmission power of

0dBm. We use OOK modulation for communication between any two devices. Since CC1310 devices are built for a maximum bandwidth of 39 KHz for OOK modulation, nodes use a 39 kHz bandwidth for transmission. The BSs use oversampling at 400 KHz bandwidth to receive the packet successfully from a node. Furthermore, BSs use a one subcarrier of 400 KHz bandwidth for inter-SNOW communication. For the experiment, we use a random payload of 10 bytes and a packet generation period of 2.45 s. For the TDMA MAC protocol, the network time is slotted, and each slot is 15 ms long. In RI-TDMA MAC, we use at most 2 re-transmission attempts for a failed transmission in the first attempt, since, we successfully received all packets with 2 re-transmission attempts for RI-TDMA during initial tests.

The experiment's default parameters are listed below.

- Number of SNOWs: 3
- Number of nodes in each SNOW: 4 - 20
- Number of Subcarrier: 28
- Subcarrier bandwidth: 400 KHz
- Traffic Pattern: Uniform
- Packet generation period of nodes: 2.45 s
- Packet Size: 10 bytes
- Transmission Power: 0 dBm

We evaluated the performance of LT-SASI using two metrics (1) maximum latency and (2) average latency. We define latency as the time between the generation of a packet at the sensor and its reception at the root-BS. We considered *maximum latency* as the longest latency experienced by a packet to reach the root BS during a 30-minute interval. We considered *average latency* as the average latency of all packets generated in the SNOW integration during a 10-minute interval. Since one node is emulating the packet transmission of many nodes, the energy measurement of the devices was not representative. Hence, we evaluated the performance of the proposed approach using energy only in simulation (Section 6). In the experiment, we evaluated the maximum latency under node scalability. We also show the maximum and average latency experienced by a packet to reach the root BS from each SNOW, to show the fairness in spectrum allocation.

Fig. 10 shows the experiment result when each node emulates the traffic of 20 devices. Under Greedy-SOP, we observed higher delay at BS1 due to high packet collisions under CSMA/CA approach. However, under LT-SASI, the observed maximum latency was almost the same for all BSs due to LT-SASI's traffic-aware subcarrier allocation to minimize the maximum latency. The experiment result in Fig. 10 shows that the spectrum allocation

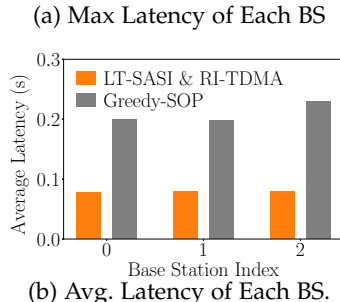
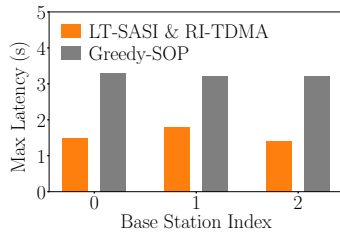


Fig. 10: Experimental Result of LT-SASI showing Latency

with LT-SASI results in similar maximum and average latencies across the BSs. Furthermore, the difference between the maximum latency of two BSs is at most 0.03 s. This result shows that LT-SASI ensures latency- and traffic-aware spectrum allocation to each BS, and fair spectrum allocation to all nodes. It also shows that LT-SASI decreases the latency by 44.3%.

Fig. 11 shows the maximum and average latency under the scalability of number of nodes. We increased the number of emulated nodes per each physical node from 5 to 20, i.e., we increased the number of nodes in the network from 60 to 300 nodes in the network. In the experiments, we observed that number of collisions increases with the increase in number of nodes and is one of the major contributors to the latency for Greedy-SOP approach. In RI-TDMA,

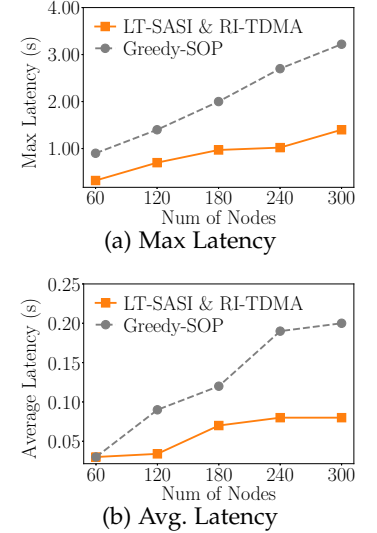


Fig. 11: Experimental Result of LT-SASI under Node Scalability

we resolved all collisions through scheduling, and hence we saved on the latency of the packet. Furthermore, we observed reduction in latency due to the subcarrier allocation through LT-SASI. The result indicates that LT-SASI results in a spectrum allocation that minimizes the maximum latency by at least 44.3% and up to 64.4%. From this result, we can conclude that LT-SASI and RI-TDMA MAC decrease the latency in the network, thereby enhancing the scalability for low-latency or real-time applications.

6 SIMULATION

We performed extensive simulations in NS3 [29] to evaluate the performance of LT-SASI with both RI-TDMA and Low-Latency TDMA MACs under large-scale integrations with many nodes and BSs and high-traffic scenarios. We integrated 5 - 25 SNOWs (depending on the simulation). We consider a random tree topology for the integrated SNOW. The root BS of the tree topology, BS_0 , is placed at the circle's center. To create connected SNOWs, we iteratively add SNOW BSs to the BSs tree topology. Upon adding a SNOW BS to the tree topology, the SNOW network is placed within the area such that the BS communicates with its parent BS. The placement of the SNOW BS is used to identify the interference with neighboring SNOWs.

Each SNOW has 100 - 1000 nodes (depending on the simulation). The nodes are randomly placed in the communication range of the BS. Each snow has 400 subcarriers of 400KHz bandwidth available for communication. The root BS computes the subcarrier allocation for different links (inter- and intra- SNOW communication) and disseminates them to all BSs. Nodes in the network periodically generate

a 30 byte packet with an interval of 32s. Since CSMA/CA MAC is prone to extremely high latencies under poor link quality, we compare the proposed approach against a best-case baseline (without any environmental conditions), demonstrating the smallest benefit of the proposed approach.

Unless specified otherwise, the default parameters for the simulation are listed below.

- Number of SNOWs: 5
- Number of nodes in each SNOW: 800
- Number of Subcarrier: 400
- Subcarrier bandwidth: 400 KHz
- Traffic Pattern: Uniform
- Packet generation period of nodes: 32 s
- Packet Size: 30 bytes
- Transmission Power: 0 dBm

In the simulations, we evaluated the performance of LT-SASI using three metrics (1) maximum latency, (2) average latency, and (3) energy consumption. For energy calculations at a node, we use the energy profile of TI CC1310 chip, which is approximately 0.128 mJ of energy consumption per one bit [32]. In the simulations, we evaluate the performance of LT-SASI under varying of number of nodes in each SNOW, varying number of BSs, and under varying number of event-triggered nodes. We simulated the network for two hours to compute the maximum latency, average latency and energy consumption of the nodes. Since the generation of an optimal solution was taking an overly long time even for a small network, we did not compare the performance of the proposed approaches with the optimal solution.

6.1 Performance under Varying Number of Nodes with Uniform Traffic

For this simulation, we considered time-triggered traffic with all nodes using a fixed period of 32 s. We also fixed the number of BSs but varied the number of nodes in a SNOW from 100 to 1000. Fig. 12 shows the results with 5 SNOWs, where we varied the total number of nodes in the network from 500 to 5000. For 500 nodes, as shown in Fig. 12a, we observed that the maximum latency of the Greedy-SOP approach was 5.4 s while that of LT-SASI & RI-TDMA and LT-SASI & TDMA were 2.04 s and 0.72 s, respectively. Similarly, as shown in Fig. 12b, we observed the average latency of the Greedy-SOP approach was 1.9 s while that of LT-SASI & RI-TDMA and LT-SASI & TDMA were 0.72 s and 0.26 s, respectively. We also observed that the maximum and average latency of the Greedy-SOP approach increases exponentially with the number of nodes, while that of LT-SASI & RI-TDMA and LT-SASI & TDMA increase linearly with the number of nodes. However, the linear increase in latencies for LT-SASI & RI-TDMA and LT-SASI & TDMA is not visible due to the steep increase in the maximum latency of Greedy-SOP approach. We observed that congestion caused at the BSs due to the allocation of only one subcarrier and collisions caused in each SNOW were the primary reasons for such high latencies in Greedy-SOP.

Fig. 12a shows that the maximum latency increases beyond 32 s (transmission period of the nodes) when the

number of nodes increases to 3500. When maximum latency increases beyond the transmission period in uniform traffic scenarios, a packet generated by a node can delay the subsequent packets generated by the same node. Such delays cause an exponential increase in Greedy-SOP's maximum and average latencies. Thus, Greedy-SOP is only scalable up to 3500 nodes for uniform traffic with 32 s periods. However, under LT-SASI & RI-TDMA and LT-SASI & TDMA the latency does not exceed 32 s even for 5000 nodes, showing extremely high scalability of the proposed integration.

To further investigate when the latency of Greedy-SOP increases beyond 32 s, we repeated the same simulation with 15 BSs. Fig. 13 shows the maximum and average latency of the proposed approaches and baseline as the number of nodes are increased from 100 to 1000. The results show that maximum latency is below 32 s when the total number of nodes in the network is less than 3000, but the maximum latency increases to 100 s as the number of nodes increases to 4500. On the contrary, the maximum latency of LT-SASI & RI-TDMA is 20 s for 15000 nodes, and that of LT-SASI & TDMA is 7 s for 15000 nodes.

The result shows that LT-SASI & RI-TDMA reduces the maximum and average latency by at least 62.2% compared to Greedy-SOP, while LT-SASI & TDMA reduces the maximum and average latency by at least 86.67%. It also shows that LT-SASI & TDMA reduces the latency by at most 66.67% when compared to LT-SASI & RI-TDMA.

The results show that the maximum and average latency for LT-SASI & RI-TDMA was higher than LT-SASI & TDMA. We observed that the extended time slot length in RI-TDMA (to accommodate the request message transmission from the BS) was the reason for the longer latency. However, TDMA MAC requires strong time synchronization between nodes. We assumed a time synchronization interval of 16 s (similar to [44]), where the BS transmits a time synchronization beacon once every 16 s, and all the nodes synchronize their clock to the BS's clock. Since listening to a packet consumes similar energy as transmitting a packet, LT-SASI & TDMA has a higher energy consumption than LT-SASI & RI-TDMA.

Fig. 12c shows the average energy consumed by a node to transmit one packet within a 2 hour interval. Under LT-SASI & RI-TDMA, a node was awake for one 40 ms time slot to listen for a request message and transmit a packet, consuming 74.08 mJ of energy. Since RI-TDMA eliminates collisions of time-triggered traffic, the energy consumption of LT-SASI & RI-TDMA was constant with an increase in the number of nodes. Under LT-SASI & TDMA, the packet transmission energy of a node was 37.044 mJ since the node was awake only for a 20ms time slot. However, the node was also awake for two additional time slots during a packet transmission period. Therefore, the average energy consumption by a node for one packet transmission is 111.12 mJ, 50% higher than LT-SASI & RI-TDMA. Note that the energy consumption remained constant for LT-SASI & TDMA since it eliminated packet collisions for time-triggered traffic. For greedy SOP, the overhead of multiple retransmissions is significantly higher than LT-SASI & RI-TDMA and LT-SASI & TDMA. We observed a similar result for 15 BSs. The results show that LT-SASI & RI-TDMA and LT-SASI & TDMA reduce the energy consumption by at

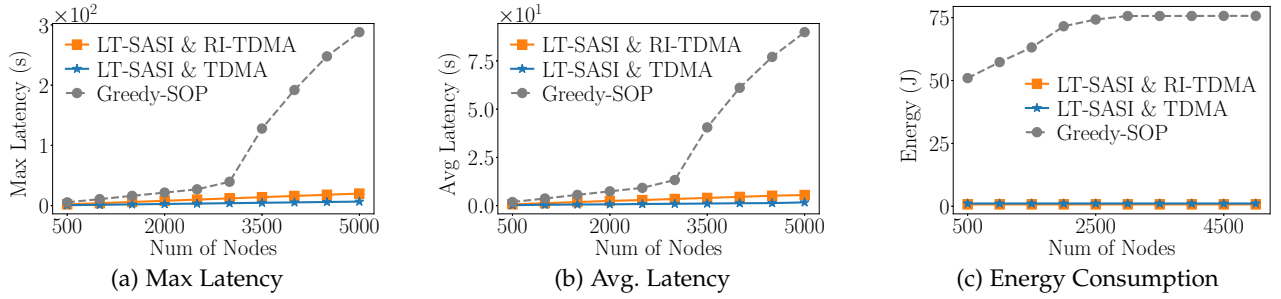


Fig. 12: Performance of LT-SASI under Varying Number of Nodes within 5 BSs and Each Node Generates Packets with a Uniform Period of 32 s

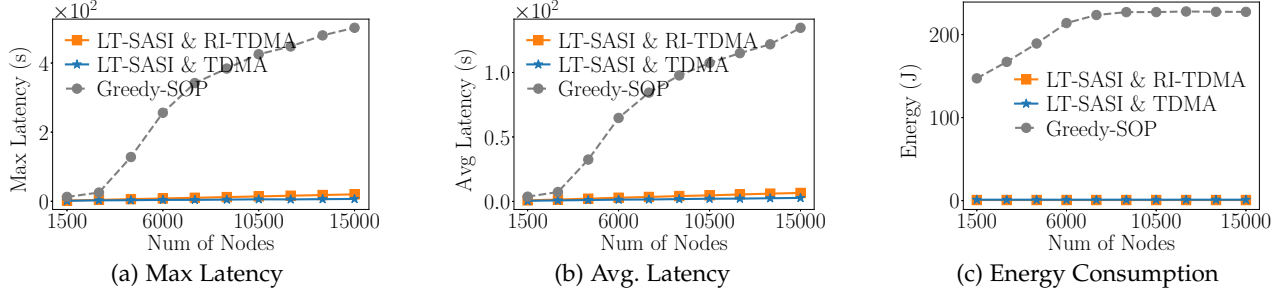


Fig. 13: Performance of LT-SASI under Varying Number of Nodes within 15 BSs and Each Node Generates Packets with a Uniform Period of 32 s

least 99% compared to Greedy-SOP.

From this result, we can conclude that the Greedy-SOP is not scalable beyond 3500 nodes for uniform traffic with 32 s periods, irrespective of the number of BSs. However, both LT-SASI & RI-TDMA and LT-SASI & TDMA are scalable beyond 15000 nodes. Additionally, LT-SASI & TDMA minimizes the maximum and average latency while LT-SASI & RI-TDMA minimizes the energy consumption at the nodes.

6.2 Performance under Varying Number of Nodes with Harmonic Traffic

For this simulation, we considered time-triggered traffic with each node generating packets with a random harmonic period in the range [16 s, 64 s]. We fixed the number of BSs but varied the number of nodes in a SNOW from 100 to 1000. Fig. 14 shows the results with 5 SNOWs, where we varied the total number of nodes in the network from 500 to 5000. For 500 nodes, as shown in Fig. 14a, we observed that the maximum latency of the Greedy-SOP approach was 7.2 s while that of LT-SASI & RI-TDMA and LT-SASI & TDMA were 1.68 s and 0.66 s, respectively. Similarly, as shown in Fig. 12b, we observed the average latency of the Greedy-SOP approach was 2.46 s while that of LT-SASI & RI-TDMA and LT-SASI & TDMA were 0.4 s and 0.17 s, respectively. We also observed that the maximum and average latency of the Greedy-SOP approach increases exponentially with the number of nodes, while that of LT-SASI & RI-TDMA and LT-SASI & TDMA increase linearly with the number of nodes. Similar to the uniform traffic simulation, we observed that congestion caused at the BSs due to the allocation of only one subcarrier and collisions caused in each SNOW were the primary reasons for such high latencies in Greedy-SOP.

We observed in Fig. 14a that the packets generated from the same node delay each other, causing an exponential increase in the maximum and average latency of the network when the number of nodes increases beyond

2500. However, under LT-SASI & RI-TDMA and LT-SASI & TDMA, the latency does not exceed 64 s (hyperperiod) even for 5000 nodes, showing extremely high scalability of the proposed integration. We repeated this simulation with 15 BSs. Fig. 15c shows the maximum and average latency of the proposed approaches and baseline as the number of nodes are increased from 100 to 1000. The results show that the maximum latency is below 64 s for 1500 nodes, but the maximum latency increases to 70 s for 3000 nodes. On the contrary, the maximum latency of LT-SASI & RI-TDMA is 16 s for 13500 nodes, and that of LT-SASI & TDMA is 7 s for 15000 nodes.

The result shows that LT-SASI & RI-TDMA reduces the maximum and average latency by at least 86% compared to Greedy-SOP, while LT-SASI & TDMA reduces the maximum and average latency by at least 90%. It also shows that LT-SASI & TDMA reduces the latency by at least 34% when compared to LT-SASI & RI-TDMA.

Fig. 12c shows the average energy consumed by a node to transmit one packet within a 2 hour interval. Under LT-SASI & RI-TDMA, a node consumed on average 148.16 mJ of energy for one packet transmission. Under LT-SASI & TDMA, a node 222.24 mJ of energy on average for one packet transmission, including energy consumed for synchronization. 50% higher than LT-SASI & RI-TDMA. For greedy SOP, the overhead of multiple retransmissions is significantly higher than LT-SASI & RI-TDMA and LT-SASI & TDMA. The results show that LT-SASI & RI-TDMA and LT-SASI & TDMA reduce the energy consumption by at least 99% compared to Greedy-SOP.

From this result, we can conclude that the Greedy-SOP is not scalable beyond 2500 nodes for traffic generated with harmonic periods in the range [16 s, 64 s], irrespective of the number of BSs. However, both LT-SASI & RI-TDMA is scalable up 13500 nodes and LT-SASI & TDMA is scalable beyond 15000 nodes. Additionally, LT-SASI & TDMA min-

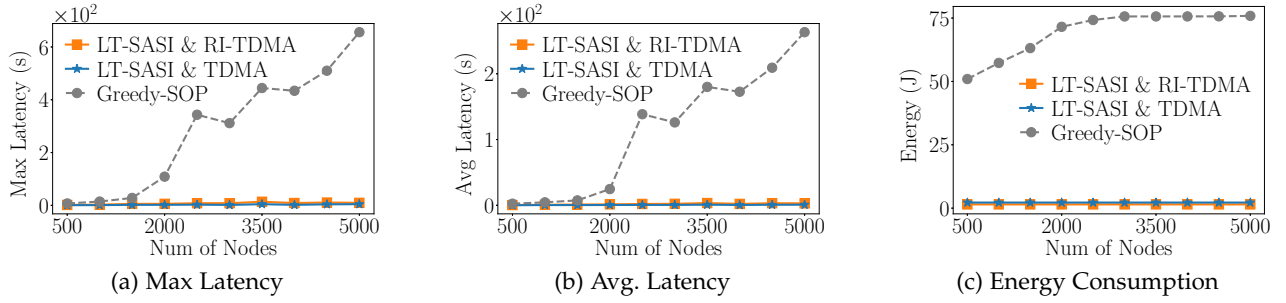


Fig. 14: Performance of LT-SASI under Varying Number of Nodes within 5 BSs and Each Node Generates Packets with Harmonic Periods in the Range [16 s, 64 s]

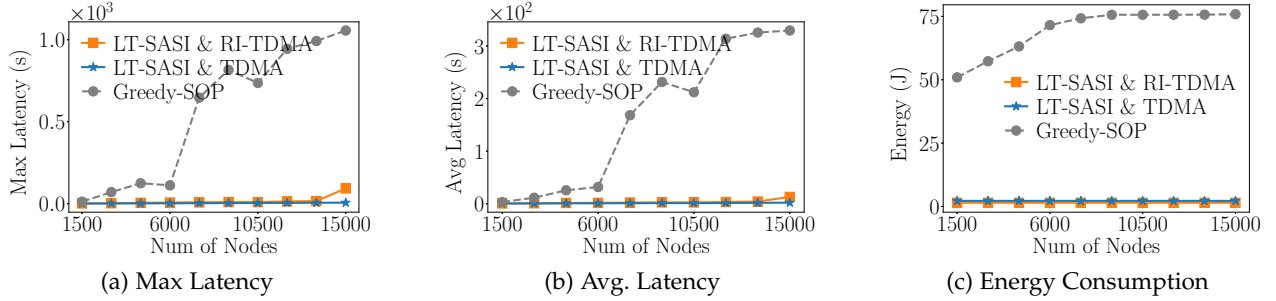


Fig. 15: Performance of LT-SASI under Varying Number of Nodes within 15 BSs and Each Node Generates Packets with Harmonic Periods in the Range [16 s, 64 s]

minimizes the maximum and average latency while LT-SASI & RI-TDMA minimizes the energy consumption at the nodes.

TDMA providing lowest maximum latency and LT-SASI & Ri-TDMA providing lower energy consumption.

6.3 Performance under Varying Number of BSs with Uniform Traffic

For this simulation, we considered time-triggered traffic with all nodes using a fixed period of 32 s. We also fixed the number of nodes in a SNOW to 800 and varied the number of BSs from 5 to 25. The total number of nodes was changed from 4000 to 20,000. In this simulation, we have observed that the maximum latency and average latency of Greedy-SOP approach increase sharply with the increase in the number of BSs. We also observed that the major source of latency was the congestion along the tree links, and the impact of such congestion was further pronounced with an increase in BSs, as shown in Fig. 16. However, in LT-SASI & RI-TDMA, the maximum and average latency were approximately 16 s and 6 s, respectively, as the number of BSs were increased. In LT-SASI & TDMA, the maximum latency increased from 5 s to 10 s as the number of BSs increased. We observed that the poor latency estimation and, thereby, insufficient subcarrier allocations along the tree link resulted in the increase in maximum latency for larger networks. However, this result shows that the proposed approach reduces the latency by 97% for LT-SASI & TDMA and LT-SASI & Ri-TDMA.

We observed that increasing the number of BSs increased the number of interfering nodes and the number of collisions in each SNOW. Increased collisions resulted in higher energy consumption in Greedy-SOP, as shown in Fig. 16c. However, the energy consumption of LT-SASI & TDMA and LT-SASI & Ri-TDMA remained consistently at 74.08 mJ and 111.12 mJ, respectively. From this result, we can conclude that LT-SASI & TDMA and LT-SASI & Ri-TDMA are scalable under the number of BSs with LT-SASI &

6.4 Performance under Varying Number of BSs with Harmonic Traffic

For this simulation, we considered time-triggered traffic with each node generating packets with a random harmonic period in the range [16 s, 64 s]. We fixed the number of nodes in a SNOW to 800 and varied the number of BSs from 5 to 25. The total number of nodes was changed from 4000 to 20,000. In this simulation, we have observed that the maximum latency and average latency of Greedy-SOP approach increase sharply with the increase in the number of BSs. We also observed that the major source of latency was the congestion along the tree links, and the impact of such congestion was further pronounced with an increase in BSs, as shown in Fig. 16. However, in LT-SASI & RI-TDMA, the maximum and average latency were approximately 16 s and 7 s, respectively, as the number of BSs were increased. In LT-SASI & TDMA, the maximum latency increased from 5 s to 8 s as the number of BSs increased. We observed that the poor latency estimation and, thereby, insufficient subcarrier allocations along the tree link resulted in the increase in maximum latency for larger networks. However, this result shows that the proposed approach reduces the latency by 97% for LT-SASI & TDMA and LT-SASI & Ri-TDMA.

We observed that increasing the number of BSs increased the number of interfering nodes and the number of collisions in each SNOW. Increased collisions resulted in higher energy consumption in Greedy-SOP, as shown in Fig. 15c. However, the energy consumption of LT-SASI & TDMA and LT-SASI & Ri-TDMA remained consistently at 74.08 mJ and 111.12 mJ, respectively. From this result, we can conclude that LT-SASI & TDMA and LT-SASI & Ri-TDMA are scalable under the number of BSs with LT-SASI &

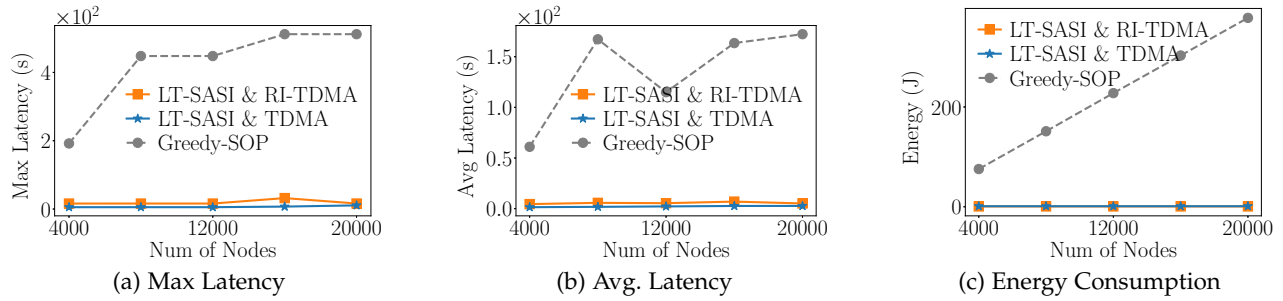


Fig. 16: Performance of LT-SASI under Varying Number of BSs and Each Node Generates Packets with a Uniform Period of 32 s

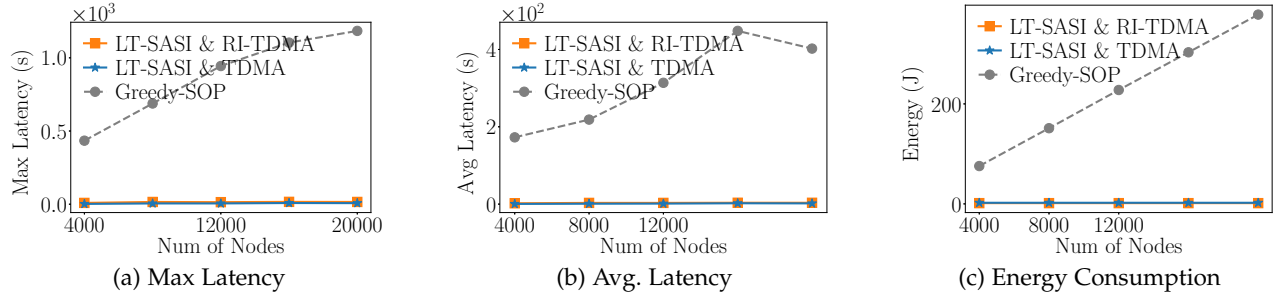


Fig. 17: Performance of LT-SASI under Varying Number of BSs and Each Node Generates Packets with Harmonic Periods in the range [16 s, 64 s]

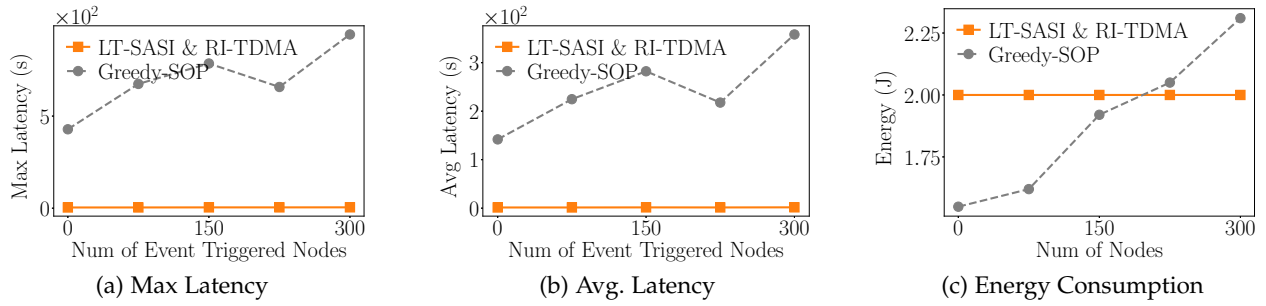


Fig. 18: Performance of LT-SASI under Varying Number of Event Triggerred Nodes

TDMA providing lowest maximum latency and LT-SASI & Ri-TDMA providing lower energy consumption.

6.5 Performance under Varying Number of Event-Triggerred Nodes

For this simulation, we generated a network with 15BSs and 100 nodes in each SNOW. We added event triggered nodes between 5–20. Each event triggered node generated packets at random intervals with period 64s or 128s. Since LT-SASI & TDMA cannot support event triggered traffic, we only compared the performance of LT-SASI & RI-TDMA with Greedy-SOP.

In RI-TDMA MAC, event triggered nodes are handled similar to time triggered node by estimating latency and assigning subcarrier based on the maximum latency. As shown in Fig. 18, increasing the number of event triggered nodes increases the maximum and average latency of the nodes. However, the increase is similar to the increase observed in time triggered nodes. Furthermore, we have observed a 99% decrease in maximum latency when compared to existing approach. From this result, we can conclude that the proposed approach can efficiently handle event-triggerred nodes.

6.6 Evaluating the Proposed Integration using Packet Reception Ratio and Throughput

For this simulation, we considered time-triggerred traffic with all nodes using a fixed period of 32 s. We also fixed the number of nodes in a SNOW to 800 and varied the number of BSs from 5 to 25. The total number of nodes was changed from 4000 to 20,000. In this simulation, we evaluate the packet reception ratio and throughput of the proposed integration under varying numbers of SNOWs. We define packet reception ratio as the ratio of the number of packets received at the root base station and the total number of packets generated by all nodes in the network. We define throughput as the number of bits successfully received at the root base station in one second.

In this simulation, we observed that the packet reception ratio of the proposed integration under RI-TDMA and TDMA MAC was at one as the number of nodes was increased, as shown in Fig. 19. RI-TDMA and

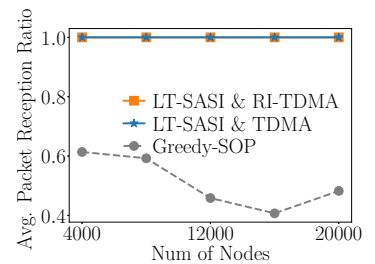


Fig. 19: Evaluating PRR of LT-SASI

TDMA MAC protocols enable a collision-free packet transmission between nodes and their SNOW BS. Furthermore, LT-SASI ensures all tree links are assigned unique subcarriers, thereby avoiding all packet collisions. For the Greedy-SOP approach, we observed that the number of collisions increased as the number of nodes increased. For this simulation, we bound the maximum number of retransmissions to 2 to ensure that packets from the same node do not interfere with each other and further reduce the PRR. Increased packet collisions and the bounded number of retransmission attempts caused the packet reception ratio of the greedy-SOP approach to decrease with the increase in the number of nodes. From this simulation, we can conclude that the proposed integration successfully receives at least 39.63% more packets than the existing approach.

In this simulation, we observed that the throughput increases of the proposed integration under RI-TDMA and TDMA MAC increased sharply as the number of nodes increased, as shown in Fig. 20.

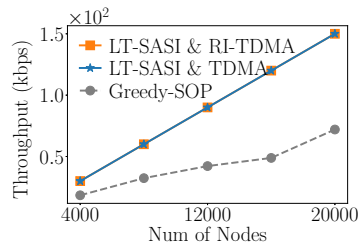


Fig. 20: Evaluating Throughput of LT-SASI

The linear increase in throughput is a result of the high PRR for the proposed integration. For the Greedy-SOP approach, we observed that the number of collisions increased as the number of nodes increased. Thus, the throughput of the Greedy-SOP increases is significantly lower than the proposed integration. This simulation concludes that the proposed integration receives at least 63.03% more bits per second than the existing approach.

7 CONCLUSION AND FUTURE WORK

LPWAN is a promising IoT technology for communicating over long distances at low power. Despite their promise, LPWANs face challenges in covering very wide areas with no/limited infrastructure. To cover such large areas, we have proposed to scale up LPWAN through seamless in-band integration of multiple SNOWs. SNOW is an LPWAN architecture over the TV white spaces. We have proposed the first latency minimizing in-band integration of multiple SNOWs for two MAC protocols, RI-TDMA MAC and TDMA MAC, assuming a single SNOW has internet access for data collection. Evaluations show both approaches outperform the existing approach. The physical experiments using RI-TDMA MAC show up to 44% reduction in the maximum network latency under our approach compared to existing approach. The simulation results show proposed approach reduces the maximum network latency at least by 62.3% under RI-TDMA MAC and 86.7% under TDMA MAC. However, SNOW integration with RI-TDMA MAC offers low energy consumption and supports a larger application base, including event-triggered traffic. Nevertheless, SNOW integration with TDMA MAC offers low latency and higher scalability for special applications requiring only time-triggered traffic.

In the future, we plan to develop an approximation algorithm for the proposed problem formulation. We plan to address the challenges arising from the mobility of SNOW nodes and energy constrained BSs. Additionally, we aim to study security challenges that arise from integrating multiple SNOWs.

REFERENCES

- [1] D. Ismail, M. Rahman, and A. Saifullah, "Low-power wide-area networks: Opportunities, challenges, and directions," in *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*, ser. Workshops ICDCN '18, 2018, pp. 8:1–8:6.
- [2] V. P. Modekurthy, D. Ismail, M. Rahman, and A. Saifullah, "Low-latency in-band integration of multiple low-power wide-area networks," in *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2021, pp. 333–346.
- [3] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra, "Low-power wide-area networks over white spaces," *ACM/IEEE Transactions on Networking*, vol. 26, no. 4, pp. 1893–1906, 2018.
- [4] FCC, Second Memorandum Opinion and Order, ET Docket No FCC 10-174, September 2010.
- [5] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with wi-fi like connectivity," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 27–38, 2009.
- [6] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2018, pp. 60–71.
- [7] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 309–321.
- [8] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications magazine*, vol. 55, no. 9, pp. 34–40, 2017.
- [9] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do lora low-power wide-area networks scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2016, pp. 59–67.
- [10] <https://www.morningagclips.com/edge-of-field-monitoring-aids-water-quality>.
- [11] "oregonFarming," 2020, <https://www.oregon.gov/ODA/shared/Documents/Publications/Administration/ORAgFactsFigures.pdf>.
- [12] <https://industrial-iot.com/2016/10/iot-oil-gas-real-power-can-provide/>.
- [13] "East texas oil field," 2020, https://en.wikipedia.org/wiki/East_Texas_Oil_Field.
- [14] "WirelessHART System Engineering Guide," 2019, http://www2.emersonprocess.com/siteadmincenter/PM%20Central%20Web%20Documents/EMR_WirelessHART_SysEngGuide.pdf.
- [15] "Emerson targets large scale deployment," 2018, <http://www.automationworld.com/networking-amp-connectivity/wireless-shines-emerson-global-user-exchange>.
- [16] "SNOW Base station," 2019, <https://github.com/snowlab12/gr-snow>.
- [17] "The future of agriculture? smart farming," <https://www.forbes.com/sites/federicoguerrini/2015/02/18/the-future-of-agriculture-smart-farming/>.
- [18] "Farming industry must embrace the internet of things vision," <http://www.beechamresearch.com/news.aspx?id=1086>.
- [19] "FarmBeats: IoT for agriculture," 2018, <https://www.microsoft.com/en-us/research/project/farmbeats-iot-agriculture/>.
- [20] "Monsanto," 2020, <https://www.rcrwireless.com/20151111/internet-of-things/agricultural-internet-of-things-promises-to-reshape-farming-tag15>.
- [21] S. Guo, F. Wang, Y. Yang, and B. Xiao, "Energy-efficient cooperative tfor simultaneous wireless information and power transfer in clustered wireless sensor networks," *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 4405–4417, 2015.

- [22] D. Zhang, H. Ge, T. Zhang, Y.-Y. Cui, X. Liu, and G. Mao, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1517–1530, 2018.
- [23] S. Liu, D.-G. Zhang, X.-h. Liu, T. Zhang, J.-x. Gao, Y.-y. Cui *et al.*, "Dynamic analysis for the average shortest path length of mobile ad hoc networks under random failure scenarios," *IEEE Access*, vol. 7, pp. 21 343–21 358, 2019.
- [24] D.-g. Zhang, S. Zhou, and Y.-m. Tang, "A low duty cycle efficient mac protocol based on self-adaption and predictive strategy," *Mobile Networks and Applications*, vol. 23, pp. 828–839, 2018.
- [25] O. Flauzac, J. Hérard, F. Nolot, and P. Cola, "A fault tolerant lora/lorawan relay protocol using lorawan class a devices," in *Ad-Hoc, Mobile, and Wireless Networks: 19th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2020, Bari, Italy, October 19–21, 2020, Proceedings 19*. Springer, 2020, pp. 295–302.
- [26] S. A. A. Kazmi, M. S. Iqbal, and S. Coleri, "Total transmission time minimization through relay selection for full-duplex wireless powered cooperative communication networks," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2020, pp. 257–268.
- [27] J. Chen, G. Mao, C. Li, and D. Zhang, "A topological approach to secure message dissemination in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 135–148, 2019.
- [28] M. Rahman and A. Saifullah, "Integrating multiple low-power wide-area networks for enhanced scalability and extended coverage," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 413–426, 2020.
- [29] "NS3," 2019, <https://www.nsnam.org>.
- [30] M. Hatler, D. Gurganious, and J. Kreegar, *Industrial LPWAN – A Market Dynamics Report*. ON World, Inc., November 2018, <https://www.onworld.com/iLPWAN/index.html>.
- [31] C.-S. Sum, M.-T. Zhou, L. Lu, R. Funada, F. Kojima, and H. Harada, "Ieee 802.15. 4m: The first low rate wireless personal area networks operating in tv white space," in *2012 18th IEEE International Conference on Networks (ICON)*. IEEE, 2012, pp. 326–332.
- [32] M. Rahman, D. Ismail, V. P. Modekurthy, and A. Saifullah, "Lpwan in the tv white spaces: A practical implementation and deployment experiences," *ACM Trans. Embed. Comput. Syst.*, vol. 20, no. 4, may 2021.
- [33] R. Prasad, *OFDM for wireless communications systems*. Artech House, 2004.
- [34] C.-T. Chen, *System and Signal Analysis*. Thomson, 1988.
- [35] "LoRaWAN," <https://www.lora-alliance.org>.
- [36] I. Katzela and M. Naghshineh, "Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey," *Commun. Surveys Tuts.*, vol. 3, no. 2, pp. 10–31, Apr. 2000.
- [37] J. Yang, M. Ding, G. Mao, Z. Lin, D.-G. Zhang, and T. H. Luan, "Optimal base station antenna downtilt in downlink cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1779–1791, 2019.
- [38] G. K. Audhya, K. Sinha, S. C. Ghosh, and B. P. Sinha, "A survey on the channel assignment problem in wireless networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 5, pp. 583–609, 2011.
- [39] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, "Spectrum assignment in cognitive radio networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1108–1135, 2013.
- [40] D.-g. Zhang, C. Chen, Y.-y. Cui, and T. Zhang, "New method of energy efficient subcarrier allocation based on evolutionary game theory," *Mobile Networks and Applications*, vol. 26, pp. 523–536, 2021.
- [41] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-end communication delay analysis in industrial wireless networks," *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1361–1374, May 2014.
- [42] Y. Chen and M. Chen, "Extended duality for nonlinear programming," *Comput. Optim. Appl.*, vol. 47, pp. 33–59, 2010.
- [43] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu, "Snow: Sensor network over white spaces," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, 2016, pp. 272–285.
- [44] V. P. Modekurthy, A. Saifullah, and S. Madria, "Distributedhart: A distributed real-time scheduling system for wireless network," in *25th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2019, pp. 216–227.

- [45] B. Andersson and J. Jonsson, "Some insights on fixed-priority preemptive non-partitioned multiprocessor scheduling," in *Proc. of the IEEE Real-Time Systems Symposium, Work-in-Progress Session*, 2000, p. 19.

[46] <https://www.ti.com/product/CC1310>.



Prashant Modekurthy is an assistant professor of computer science at University of Nevada, Las Vegas. He received his PhD in computer science from Wayne State University in 2020. His research interests are in real-time systems, cyber physical systems, Internet-of-Things, and wireless networks. He received one best paper award in IEEE ICII 2018.



Dali Ismail is an assistant professor in the Department of Computer Science at Binghamton University. He received his PhD in computer science from Wayne State University in 2021 and his MS from Washington University in St. Louis. His research concerns the Internet of Things, mobile and wireless networks, and embedded and real-time systems. He received three best paper awards/nominations in highly competitive conferences, including ACM SenSys (2016 nomination), IEEE ICII (2018 award), and ACM EWSN (2021 nomination). He also received an outstanding teaching award from SIUE School of Engineering (2023).



Mahbubur Rahman is an assistant professor of computer science at The City University of New York - Graduate Center and Queens College. He received his PhD in computer science from Wayne State University in 2020. His research spans a variety of topics in wireless networks and communication, embedded and distributed systems, and embedded learning and security. He received six best paper awards and nominations from several conferences and organizations.



Abusayeed Saifullah is an associate professor of Computer Science at Wayne State University. He received PhD in Computer Science and Engineering with Turner Dissertation Award from Washington University in St Louis. His research concerns Internet-of-Things (IoT), real-time and embedded systems, cyber-physical systems, and wireless networks. Dr. Saifullah received ten Best Paper Awards and nominations in highly competitive conferences including IEEE RTSS (2019, 2014, 2011), ACM SenSys (2016 nomination), IEEE ICII (2018), ACM EWSN (2021 nomination), and IEEE RTAS (2012 nomination). He has a total citations of over 6,000 with an h-index of 33 according to Google Scholar. He is a recipient of the NSF CAREER award (2019) and Wayne State University's College of Engineering Faculty Research Excellence Award (2020). He is an associate editor of ACM Transactions on Cyber-Physical Systems and an area editor of Elsevier *Pervasive and Mobile Computing*.