2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS) | 978-1-6654-0386-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/RTAS52030.2021.00034

Low-Latency In-Band Integration of Multiple Low-Power Wide-Area Networks

Venkata P. Modekurthy*, Dali Ismail,†, Mahbubur Rahman‡, Abusayeed Saifullah†
*University of Nevada Las Vegas, NV, USA
†Wayne State University, MI, USA
‡Queens College, City University of New York, NY, USA

***Abstract*—Today, 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, specially in rural or remote areas with no/limited infrastructure, requiring an in-band integration of multiple LPWANs. To avoid the crowd in the limited ISM band and the cost of licensed band and infrastructure, *SNOW (Sensor Network Over White spaces)* is an LPWAN architecture over the TV white spaces. It offers high scalability through concurrent and bi-directional communication between a base station and numerous nodes. We consider a seamless integration of multiple SNOWs. Existing approach does not consider minimizing network latency and is less suitable for delay-sensitive or real-time applications. We propose the first scalable in-band integration of multiple SNOWs that minimizes network latency. By taking into account the impact of bandwidth on latency and base station power dissipation, we formulate low-latency integration of multiple SNOWs as a constrained spectrum allocation problem. It is solved through a greedy algorithm by analyzing network latency and by adopting a latency- and traffic- aware bandwidth allocation along the links to achieve an integrated network. We have implemented the proposed integration both on SNOW hardware and in NS-3 simulator. Both physical experiments and simulations show a significant reduction (44% and 97%, resp.) in network latency under our approach compared to existing approach.**

I. 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], [2]. With the fast growth of IoT, multiple LPWAN technologies have recently emerged such as LoRa [3], SigFox [4], IQRF [5], RPMA [6], DASH7 [7], Weightless-N/P [8], Telensa [9] in the ISM band, and EC-GSM-IoT [10], NB-IoT [11], LTE Cat M1 [12], [13] in the licensed cellular band. To avoid the *crowd* in the *limited* ISM band and the *cost* of licensed band, *SNOW (Sensor Network Over White spaces)* is an LPWAN architecture to support scalable wide-area IoT over the TV white spaces [14]–[16]. *White spaces* are the allocated but locally unused TV spectrums where unlicensed devices can operate as secondary users [17], [18]. 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 [18]. 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 [19].

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 [20]–[25]. Today, industrial and agricultural IoT and cyber-physical systems are emerging in large-scale and wide-area applications. Specifically, agricultural fields [26]–[28] and oil/gas fields [29] 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 [30]. Emerson is targeting to deploy 10,000 nodes for managing an oil-field in Texas [31], [32]. 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 no/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 [14]–[16]. Its implementation is available as open-source [33]. 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 [34], [35]. 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 [36]–[38]. Industries like Microsoft [39], [40] and Monsanto [41] are hence promoting agricultural IoT.

978-1-6654-0386-3/21/$31.00 ©2021 IEEE
DOI 10.1109/RTAS52030.2021.00034

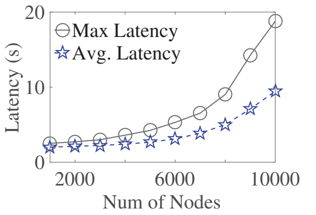


Fig. 1: Latency under Existing SNOW Integration.

In-band integration considered in this paper is conceptually different from and more challenging than traditional tiered or clustered wireless sensor network (WSN) [42] or 802.11 mesh [43]. Unlike these networks, integrating multiple SNOWs needs to find proper bandwidths (i.e., the physical width of the spectrum or band) for all links which are inter-dependent and subject to BS power dissipation. Besides, 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. While a seamless integration of SNOWs was studied in a recent work [44], its objective is to trade the scale (in terms of number of total nodes) for inter-SNOW interference. Its bandwidth allocation for inter-SNOW communication does not take into account the link traffics. As a result, it performs poorly in terms of network latency, eventually affecting the scale. A simulation using NS-3 [45] in such an integrated network of five SNOWs shows that the maximum latency for collecting a packet increases exponentially and deviates sharply from average latency with the number of nodes (Fig. 1). 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% of industrial IoT professionals are targeting LPWAN for industrial WSN applications [46], [47].

We propose the first scalable in-band integration of multiple SNOWs that minimizes network latency. In each SNOW, we adopt a new media access control (MAC) protocol 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.
- The formulated problem is solved through a greedy algorithm 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 other future LPWANs in white spaces (e.g., based on upcoming 802.15.4m standard [48]).
- 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 [44]. We have also performed simulations in NS-3 [45] 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 II presents an overview of SNOW. Section III gives an overview of the proposed integration along with related work, challenges, and problem formulation. Section IV details the solution for latency minimizing integration. Section V, VI, and VII present experiments, simulations, and conclusion respectively.

II. The SNOW Architecture

SNOW is a promising LPWAN technology with open-source implementation on commercially available off-the-shelf devices [33]. 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 [14]. 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 [49].

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 ($f_1, f_2, \cdots$), each of equal bandwidth. Note that, ***bandwidth*** 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 on a single subcarrier, as shown in Fig. 2.

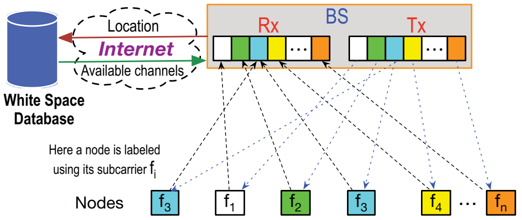


Fig. 2: SNOW architecture with dual radio BS and subcarriers [14].

To enable many simultaneous transmissions/receptions, the physical layer is designed based on a **D**istributed 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, or so depending on packet size and needed bit rate). The nodes transmit/receive on orthogonal subcarriers, each using one. 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 using one antenna. Similarly, it can transmit different data on different subcarriers through a single transmission.

Both radios of the BS use the same bandwidth and subcarriers - 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 like the one used in TinyOS [50]. In this paper, we adopt a time-triggered MAC protocol that blends with the proposed low-latency integration.

III. Integrating Multiple SNOWs: An Overview of the Proposed Approach

A. 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 [15]. 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 commercially available 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, $\cdots$ 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 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 [48]. 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. For example, in Fig. 3, BS_2 collects information from nodes in SNOW_2 and relays it to the root BS via the tree links $\text{BS}_2 \rightarrow \text{BS}_1 \rightarrow \text{BS}_0$. Scalability through in-band integration of multiple SNOWs impacts the delivery *latency*, and hence, we propose to minimize such latency.

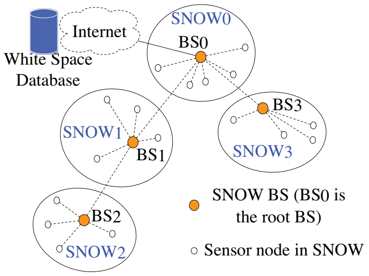


Fig. 3: Integrating Multiple SNOWs to Build a SNOW-tree.

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 [51], [52]. 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 multi-objective (latency and PAPR) of our approach, we will minimize latency while keeping PAPR as a constraint.

B. Related Work and New Challenges

To cover a wide area, LoRa integrates multiple gateways through the Internet [3]. Cellular networks do the same relying on wired infrastructure [53]. Rural and remote areas lack such infrastructure. Our proposed in-band integration is conceptually different from traditional tiered or clustered WSN [42] or 802.11 mesh [43]. 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 [54], [55]) or cognitive radio networks [56]–[58] cannot be used for bandwidth allocation in SNOW integration.

While SNOW integration was studied in a recent work [44], 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 assigns subcarriers to each SNOW of the integrated network. The approach 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 network performance. As a result, it performs poorly in terms of network latency, eventually affecting the scale. In contrast, we propose the first scalable in-band integration of multiple SNOWs that minimizes network latency. This is done through a traffic-aware bandwidth allocation to the links so as to minimize the maximum latency, and by adopting a deterministic MAC protocol that blends with the integration.

C. Formulation

We want to minimize the maximum latency among all SNOWs for data delivery to root base station BS_0. Considering a uniform subcarrier bandwidth and spacing across all SNOWs, let Z_i be the set of orthogonal subcarriers available at BS_i. Let V_i be the set of nodes in SNOW_i and $n_i = |V_i|$. Each node $u \in V_i$ generates packets periodically with a period $T_i(u)$. Time needed for one packet transmission is considered a time unit. A subset $S_i \subset Z_i$ will be assigned for SNOW_i for its nodes V_i to deliver their data to BS_i. Let $\rho(i) \in \{0, 1, \cdots, N-1\}$ be such that $\text{BS}_{\rho(i)}$ is the parent of BS_i in the tree. The subcarriers allocated for communication along tree link $\text{BS}_i \rightarrow \text{BS}_j$, where BS_j is the parent of BS_i, is denoted by set $S_{i,j}$. Thanks to the capability of D-OFDM to enable parallel data streams to/from numerous distributed nodes from/to the BS, the proposed integration becomes *seamless* meaning that inter-SNOW and intra-SNOW communication can happen in parallel. To achieve this, subcarriers allocated on the tree link should not overlap with those for intra-SNOW communication, i.e., $S_{i,j} \subset Z_i$ and $S_{i,j} \subset Z_j$, and $S_{i,j} \cap S_i = S_{i,j} \cap S_j = \emptyset$.

Let $A(i), D(i) \subseteq \{0, 1, \cdots, N-1\}$ be such that each $B_k, k \in A(i)$, is an ancestor and each $B_j, j \in D(i)$, is a descendant of BS_i in the SNOW-tree. For a packet of u of SNOW_i, let $\lambda(u, u, \text{BS}_i)$ 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 $\lambda(u, \text{BS}_j, \text{BS}_{\rho(j)})$ be its estimated latency along tree link $\text{BS}_j \rightarrow \text{BS}_{\rho(j)}$. Then, its total estimated latency $\Lambda(u)$ is given by Equation (1).

$$\Lambda(u) = \lambda(u, u, \text{BS}_i) + \sum_{j \in A(i)-\{0\}} \lambda(u, \text{BS}_j, \text{BS}_{\rho(j)}) \qquad (1)$$

Let L_i denote the maximum $\Lambda(u)$ in SNOW_i. That is,

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

Note that $L(i)$ is an estimate of maximum latency for collecting a packet at root base station BS_0 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 this metric.

If some communication in SNOW_i is interfered by another in SNOW_j, then SNOW_j is its ***interferer.*** Let $I_i \subset \{0, 1, \cdots, N-1\}$ be such that each $\text{SNOW}_j, j \in I_i$, is an interferer of SNOW_i. Let $J_i \subset \{0, 1, \cdots, N-1\}$ be such that a transmission along link $\text{BS}_i \rightarrow \text{BS}_{\rho(i)}$ can be interfered by that along $\text{BS}_k \rightarrow \text{BS}_{\rho(k)}$ or by some node's transmission in $\text{SNOW}_k, k \in J_i$. In the following, Constraints (3) and (4) 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 define any allowed spectrum overlap between two interfering SNOWs where $0 \leq \sigma_i \leq 1$. Let β_i be the maximum number of subcarriers on which BS_i can transmit simultaneously having 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 for $0 \leq i < N$ and $S_{i,\rho(i)}$

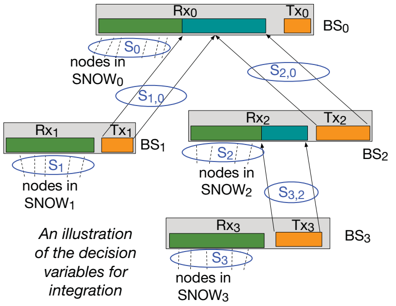


Fig. 4: Integration using BS dual-radios.

for $0 < i < N$ so as to

$$\begin{aligned} \text{Minimize} \quad & \max\{L(i) | 0 \le i < N\} \\ \text{subject to} \quad & S_i \subseteq Z_i, \quad S_{i,\rho(i)} \subseteq Z_i, \quad S_{i,\rho(i)} \subseteq Z_{\rho(i)} && (2) \\ & S_{i,\rho(i)} \cap S_j = \emptyset, \forall j \in I_{\rho(i)} \cup \{i, \rho(i)\} && (3) \\ & S_{i,\rho(i)} \cap S_{j,\rho(j)} = \emptyset, \forall j \in J_i \cup \{\rho(i)\} && (4) \\ & \sum_{j \in I_i} |S_i \cap S_j| \le \sigma_i |S_i|; \quad 1 \le |S_{i,\rho(i)}| < \beta_i && (5) \end{aligned}$$

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. Furthermore, the above formulation is applicable for any MAC protocol used inside each SNOW. However, network latency expressions usually take after multi-processor response-time analysis and become non-linear and often non-differentiable, and so does the above optimization problem [59], [60]. Adopting a subgradient approach or some global optimization framework such as Simulated Annealing based penalty method [61] for an optimal solution can be extremely time consuming, especially due to spectrum dynamics of white spaces, which would require a frequent re-run of the optimization. Instead, we propose a highly intuitive greedy heuristic that provides a fast heuristic 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. Such subcarriers are identified based on a latency expression. Hence, we first describe each SNOW's MAC protocol and derive its latency expression, and then describe the spectrum allocation algorithm.

IV. Description of the Low-Latency Integration of Multiple SNOWs

Since carrier sensing based MAC protocols (e.g., CSMA/CA) may not provide predictable latency, they are less preferred in time-sensitive applications. Thus, we choose to use a time-triggered MAC protocol based on TDMA (time division multiple access) inside each SNOW. However, TDMA based protocols require a strong time synchronization, which consumes high energy at the nodes. Furthermore, such protocols require a central manager to compute and disseminate a schedule to all 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 [62].

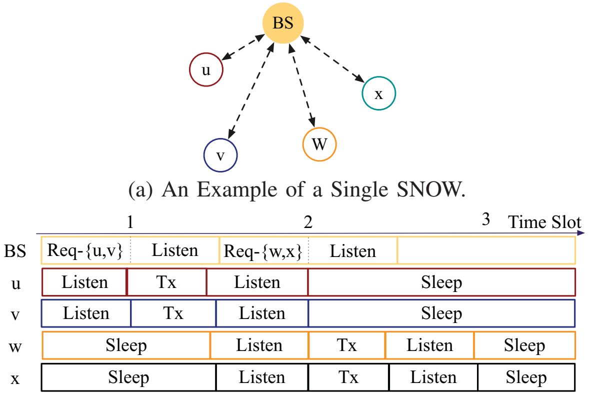
To overcome the above limitations, we propose to adopt a receiver-initiated MAC for TDMA, RI-TDMA, where the BS schedules nodes for transmission in each slot via a request message. A receiver-initiated message eliminates the need for executing a time synchronization protocol at the node. Note that any TDMA based MAC protocol with good latency estimation can be used. However, we choose RI-TDMA MAC since it is scalable and handles transmission failures and packet re-transmissions dynamically.

A. *RI-TDMA MAC for SNOW*

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 $1, 3, 5, 7, \cdots$ and BS$_1$ will use the subcarrier for time slots $2, 4, 6, 8, \cdots$. To determine the time slots with uninterrupted access to a common subcarrier,



(a) An Example of a Single SNOW.

(b) Transmission Schedule for the Example Network.

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

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 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.

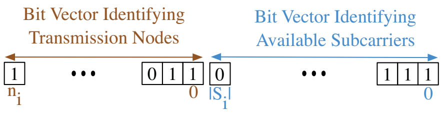


Fig. 6: Payload from BS$_i$'s Request Message.

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 [44].

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 sucarrier 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.

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 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.

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. 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. 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 on the next available subcarrier and time slot.

Similar to intra-SNOW communication, packets in inter-SNOW communication are prioritized based on the RM scheduling policy and scheduled for transmission locally by a BS. A BS locally determines the next available packets for transmission based on the packets' priority in its queue. Note that, unlike the SNOW nodes, the BS has sufficient energy and memory. Furthermore, a packet generated by an event-triggered node has a lower priority compared to a packet generated by a time-triggered node.

B. Latency Estimation in 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$_i$, $\lambda(u, u, \mathrm{BS}_i)$. 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 $v, v \in \mathrm{SNOW}_i$. 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 $\lambda(u, u, \mathrm{BS}_i)$ is given by the following expression:

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

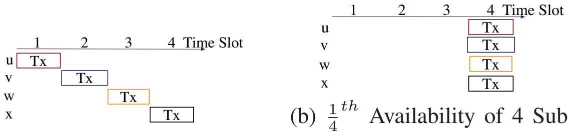
Each packet of a high priority node delays the packet of node u by one time slot.

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 shows below

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

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).



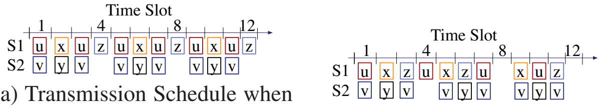
(a) Continuous Availability

(b) $\frac{1}{4}^{th}$ Availability of 4 Subcarriers

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

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 (6) gives the maximum latency experienced by a packet from node u.

$$\lambda(u, u, \text{BS}_i) = 1 + \varsigma + \left\lceil \frac{1}{\psi_i} \sum_{v \in hp_i(u)} \left\lceil \frac{\lambda(u, u, \text{BS}_i)}{T_i(v)} \right\rceil \right\rceil \tag{6}$$



(a) Transmission Schedule when $T_u = T_v = 2$ and $T_x = T_y = T_z = 4$

(b) Transmission Schedule when T_u changes to 3

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

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 [63] 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 (6) 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 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:

$$\lambda(u, u, \text{BS}_i) \left\lceil \frac{\psi_i}{2\epsilon |S_{i,\rho(i)}|} \right\rceil$$

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 transmits 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 $C(i) \subseteq \{0, 1, \cdots, N-1\}$ 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 $\lambda(u, \text{BS}_i, \text{BS}_{\rho(i)})$ is given by the following Equation:

$$\left\lceil \sum_{\substack{v \in hp_g(u) \,\&\, g \in D(k) \\ k \in C(i)}} \frac{\lambda(u, \text{BS}_i, \text{BS}_{\rho(i)})}{T_g(v)\, |S_{k,i}|} + \sum_{w \in hp_i(u)} \frac{\lambda(u, \text{BS}_i, \text{BS}_{\rho(i)})}{T_i(w)\, \psi_i} \right\rceil$$

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 (7).

$$\phi_{i,\rho(i)} = \min(2\epsilon|S_{i,\rho(i)}|\,,\, \psi_i + \sum_{k\in C(i)} |S_{k,i}|) \tag{7}$$

Thus, the maximum latency experienced by a packet of u to reach $\mathrm{BS}_{\rho(i)}$ from BS_i is given by the sum of two sources. The solution to the expression in Equation (8) gives the maximum latency experienced by a packet from u at BS_i.

$$\lambda(u, \mathrm{BS}_i, \mathrm{BS}_{\rho(i)}) = \lambda(u, u, \mathrm{BS}_i)\left\lceil \frac{\psi_i}{2\epsilon|S_{i,\rho(i)}|} \right\rceil + \left\lceil \frac{1}{\phi_{i,\rho(i)}} \left\lceil \sum_{\substack{v\in hp_g(u)\,\&\, g\in D(k)\\ k\in C(i)}} \frac{\lambda(u, \mathrm{BS}_i, \mathrm{BS}_{\rho(i)})}{T_g(v)\,|S_{k,i}|} + \sum_{w\in hp_i(u)} \frac{\lambda(u, \mathrm{BS}_i, \mathrm{BS}_{\rho(i)})}{T_i(w)\,\psi_i} \right\rceil \right\rceil \tag{8}$$

The maximum latency experienced by a packet from u at BS_j (the parent of BS_i) is given by the following expression.

$$\lambda(u, \mathrm{BS}_j, \mathrm{BS}_{\rho(j)}) = \lambda(u, \mathrm{BS}_i, \mathrm{BS}_j)\left\lceil \frac{\psi_i}{2\epsilon|S_{i,\rho(j)}|} \right\rceil + \left\lceil \frac{1}{\phi_{j,\rho(j)}} \left\lceil \sum_{\substack{v\in hp_g(u)\,\&\, g\in D(k)\\ k\in C(j)}} \frac{\lambda(u, \mathrm{BS}_j, \mathrm{BS}_{\rho(j)})}{T_g(v)\,|S_{k,j}|} + \sum_{w\in hp_j(u)} \frac{\lambda(u, \mathrm{BS}_j, \mathrm{BS}_{\rho(j)})}{T_j(w)\,\psi_j} \right\rceil \right\rceil$$

Note that the above expression is derived from Equation (8) by replacing the transmission from node BS_i to BS_j with the transmission from BS_j to $\mathrm{BS}_\rho(j)$.

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:

$$\Lambda(u) = \lambda(u, u, \mathrm{BS}_i) + \sum_{j\in A(i)-\{0\}} \lambda(u, \mathrm{BS}_j, \mathrm{BS}_{\rho(j)}).$$

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 above latency formulation can be used to generate a solution for the bandwidth allocation problem formulation in Section III-C. 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.

Algorithm 1: LT-SASI

```
Input  : Tree Structure, $Z_i, I_i, J_i, \forall i \in \{0, N\}$
         $T_u \forall\, u \in \mathrm{BS}_i,\ 0 \le i < N$
Output: $S = \cup_{\forall i}\{S_i, S_{i,\rho(i)}\}$
1  for $i \in \{0, N\}$ do
2      $S_i$ = Unique_Subcarrier($Z_i, I_i, S$)
3      $S_{i,\rho(i)}$ = Unique_Subcarrier($Z_i, Z_{\rho(i)}, J_i, S$)
4  end
5  do
6      $L$ = Compute_Latency()
7      $A$ = Exclude_infeasible_links($L, S$)
8      if $|A| = 0$ then
9          break
10     end
11     $v, \nu$ = Identify_Bottleneck_Link($A$)
12     if $v \in BS$ then
13         $S_{v,\nu}$ = Add_Feasible_Channel($S, Z$)
14     else
15         $S_\nu$ = Add_Feasible_Channel($S, Z$)
16     end
17 while True
```

C. 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 left that minimize the 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

(2), (3), (4), and (5). 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 ee-dissemination of subcarrier information obviates the need for a global agreement.

The time complexity of the `Unique_Subcarrier` function is $O(|Z|)$, where $|Z|$ represents the maximum number of subcarriers available in a SNOW. Iterating over N SNOW BSs, the time complexity for the initial subcarrier assignment is $O(N|Z|)$. The time complexity of computing the maximum latency for all nodes is $O(N|V|)$, where $|V|$ represents the maximum number of nodes in a SNOW. The time complexity of `Exclude_infeasible_links` function is $O(N|Z|)$, considering that function iteratively tries to find a feasible subcarrier for each link. The time complexity of identifying the bottleneck link is $O(N)$ since the maximum length of the tree structure is N. The time complexity of a brute force algorithm to find an available subcarrier that minimizes the latency of the bottleneck link is $O(|Z|)$. Iterating over $N|Z|$ times to exhaustively search through all aspirant links, the time complexity of LT-SASI algorithm is $O(N|Z| + N|Z|(N + N|V| + N|Z|)) = O(N^2|V||Z| + N^2|Z|^2)$.

V. Experiment

Here, we evaluate the proposed low-latency integration of SNOWs through real experiments. We compare the performance with the greedy heuristic for scalability optimization (SOP) proposed in [44], named *Greedy-SOP* here. Note that, in this paper, we have adopted a new MAC protocol, RI-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, for the sake of originality of the baseline approach, we cannot compare the two integration approaches considering the same MAC protocol. 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 approach with the optimal solution.

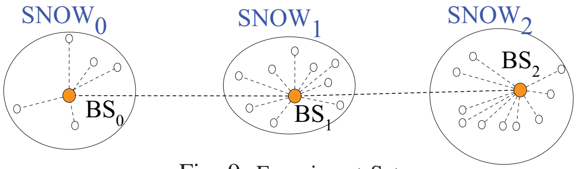
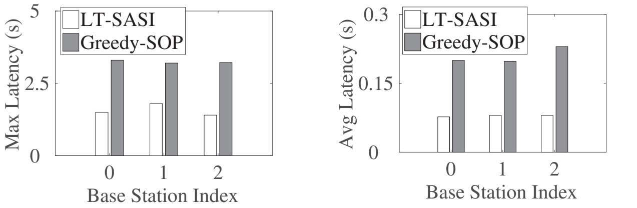


Fig. 9: Experiment Setup

We create a SNOW integration with 3BSs, as shown in Fig. 9. We were unable to perform large scale outdoor experiments due to safety hazards and restricted operation in the campus under the surging COVID-19 pandemic. However, we performed large scale evaluation with a wide range of scenarios considering random tree structures in the simulations (Section VI). In the experiment, each BS uses 2 USRP devices –one for Tx radio and the other for Rx radio. We have adopted the opensource implementation of SNOW that is available at [33]. Each SNOW network is assigned 5 CC1310 devices [65]. 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.

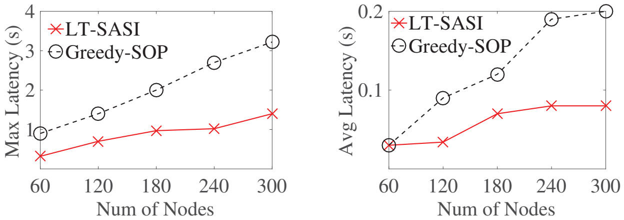
We used 28 subcarriers from 2 TV channels in the range of 500MHz to 512MHz of the available white space spectrum. Both CC1310 and USRP devices use a transmission power of 0dBm. We use OOK modulation for communication between any two devices. Since CC1310 devices are built for a maximum bandwidth of 39KHz for OOK modulation, nodes use a 39kHz bandwidth for transmission. The BSs use oversampling at 400KHz bandwidth to receive the packet successfully from a node. Furthermore, BSs use a 400KHz bandwidth for inter-SNOW communication. For the experiment, we use a random payload of 10bytes and a packet generation interval of 2.5s. For the TDMA MAC protocol, the network time is slotted, and each slot is 15ms long. In RI-TDMA MAC, we use at most 2 re-transmission attempts for a failed transmission in the first attempt.

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 VI). 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.



(a) Max Latency of Each BS (b) Average Latency of Each BS.

Fig. 10: Experimental Result of LT-SASI showing Maximum and Average Latency to Reach the BS$_0$ from Each SNOW

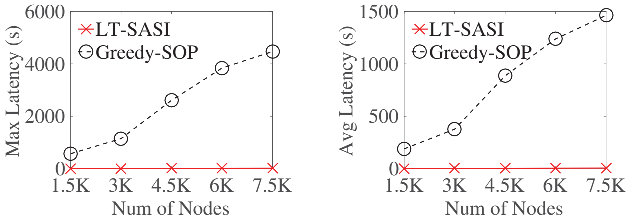


(a) Max Latency (b) Average Latency

Fig. 11: Experimental Result of LT-SASI under Varying Number of Nodes

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 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 0.03s. This result shows that LT-SASI ensures latency- and traffic-aware spectrum allocation to each BS, and fair spectrum allocation to all nodes in the network. It also demonstrates that LT-SASI algorithm 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, 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.



(a) Max Latency (b) Average Latency

Fig. 12: Performance of LT-SASI under Varying Number of Nodes

VI. Simulation

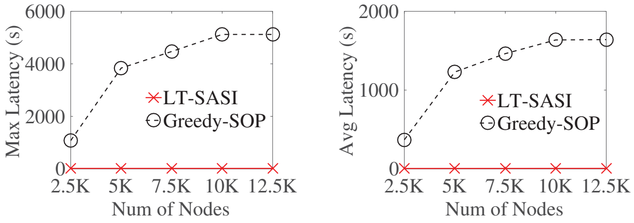
We performed extensive simulations in NS3 [45] to evaluate the performance of LT-SASI under large scale integrations with many nodes and BS. In the simulation, we used a SNOW integration of 15 SNOWs. Each SNOW consists of 100 nodes. We consider a random tree topology for the integrated SNOW. We assumed 200 subcarriers of 400KHz bandwidth are available for communication at each SNOW. 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. Unless specified otherwise, these are our default parameter settings.

In the simulations, we evaluated the performance of LT-SASI using 3metrics (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 50mW of power consumption [65]. 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.

A. Performance under Varying Number of Nodes

For this simulation, we varied the number of nodes in a SNOW from 100 to 500, i.e., we varied the total number of nodes in the network from 1500 to 7500. In this simulation, we observed that the maximum and average latency for transmitting packets is significantly larger for the Greedy-SOP approach. We observed that allocating one subcarrier along the tree links was the primary reason for such high latencies in Greedy-SOP. In LT-SASI, we minimize the latency by allocating multiple subcarriers to high traffic links. For this reason, we observed that the proposed approach reduces the average and maximum latency by at least 99%, as shown in Fig. 12. Note that, there was a steady increase in the maximum latency from 3s to 19s as the number of nodes are increased for LT-SASI. However, this increase is not visible due to the steep increase in the maximum latency of Greedy-SOP approach.

In this simulation, we also compared the energy consumption of the proposed approach against Greedy-SOP approach. We used average energy consumption for packet transmissions at SNOW nodes as a metric for comparison. Under LT-SASI, the average number of packet transmissions was close to 1 and did not change with the number of nodes. Thus, the average



(a) Max Latency

(b) Average Latency

Fig. 14: Performance of LT-SASI under Varying Number of BSs

energy consumption remained constant. The reception of request messages from a BS introduces energy overhead at each node in RI-TDMA MAC. For a network with only 1500 nodes, this energy overhead in RI-TDMA is higher than the energy overhead due to multiple retransmissions in CSMA/CA, as shown in Fig. 13. Nevertheless, for a network with more than 3000 nodes, the energy overhead due to collisions in Greedy-SOP is significantly higher than the energy overhead of RI-TDMA MAC. From this result, we can conclude that LT-SASI minimizes the maximum and average latency while consuming similar or less energy than Greedy-SOP.

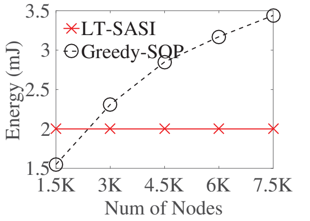


Fig. 13: Energy Consumption of LT-SASI under Varying Number of Nodes

B. Performance under Varying Number of BSs

For this simulation, we kept the number of nodes in a SNOW at 500 and varied the number of BSs from 5 to 25, i.e., the total number of nodes were varied from 2500 to 12500. In this simulation, we have observed that the maximum latency and average latency of Greedy-SOP approach increases 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. 14. However, in LT-SASI, the maximum latency and average latency remain consistent at 19s and 6s, respectively, as the number of BSs were increased. We observed that LT-SASI approach balances the amount of inter-snow traffic at each BSs, thereby reducing the impact of the number of BSs on the network. This result shows that the proposed approach reduces the latency by 97%.

We observed that increasing the number of BSs increased the number of interfering nodes and the number of collisions in each SNOW. Increased number of collisions resulted in higher energy consumption in Greedy-SOP, as shown in Fig. 15. Since a BS resolves all transmission conflicts in RI-TDM, an increase in the number of BS the number of transmissions made by a node. Thus, the energy consumed by a packet is consistent as the number of BSs were increased. From this result, we can conclude that the proposed approach is scalable under the number of BSs and minimizes the network's latency while consuming less energy than existing approach.

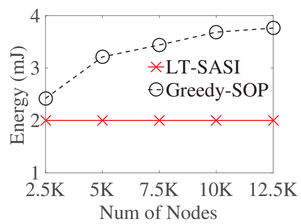
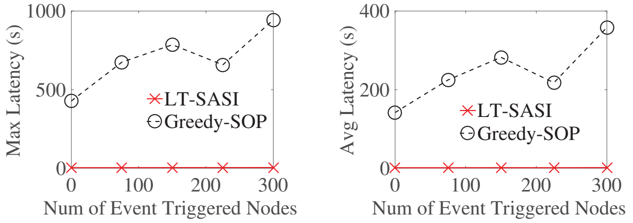


Fig. 15: Energy Consumption of LT-SASI under Varying Number of BSs



(a) Max Latency

(b) Average Latency

Fig. 16: Performance of LT-SASI under Varying Number of Event Triggered Nodes

C. Performance under Varying Number of Event-Triggered 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. 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. 16, 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-triggered nodes.

VII. Conclusion

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 for IoT applications in agriculture, oil, and gas fields specially in rural/remote areas with no/limited infrastructure. To cover such large-areas, we have proposed to scale up LPWAN through a 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. Both physical experiments and simulations show a significant reduction (50% and 97%, respectively) in network latency under our approach compared to existing approach. In the future, we plan to address the challenges arising from mobility of SNOW nodes.

VIII. Acknowledgment

This work was supported by NSF through grants CAREER-1846126 and CNS-2006467.

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] H. H. R. Sherazi, L. A. Grieco, M. A. Imran, and G. Boggia, "Energy-efficient lorawan for industry 4.0 applications," *IEEE Transactions on Industrial Informatics*, pp. 1–11, 2020.

[3] "LoRaWAN," https://www.lora-alliance.org.

[4] "SIGFOX," 2020, http://sigfox.com.

[5] "IQRF," 2020, http://www.iqrf.org/technology.

[6] https://www.ingenu.com/technology/rpma.

[7] "Dash7," 2020, http://www.dash7-alliance.org.

[8] "Weightless," 2020, http://www.weightless.org.

[9] "Telensa," 2020, https://www.telensa.com.

[10] "EC-GSM-IoT," 2020, https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf.

[11] "NB-IoT," 2020, https://www.u-blox.com/en/narrowband-iot-nb-iot.

[12] https://www.u-blox.com/en/lte-cat-m1.

[13] "LTE Advanced Pro," 2017, https://www.qualcomm.com/invention/technologies/lte/advanced-pro.

[14] 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.

[15] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu, "SNOW: Sensor network over white spaces," in *SenSys '16 (Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems)*. ACM, 2016, pp. 272–285.

[16] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra, "Enabling reliable, asynchronous, and bidirectional communication in sensor networks over white spaces," in *15th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2017, pp. 1–14.

[17] FCC, ET Docket No FCC 08-260, November 2008.

[18] FCC, Second Memorandum Opinion and Order, ET Docket No FCC 10-174, September 2010.

[19] 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.

[20] 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.

[21] 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.

[22] O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.

[23] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range and amp; low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016.

[24] 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.

[25] 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.

[26] https://www.morningagclips.com/edge-of-field-monitoring-aids-water-quality.

[27] https://cees.iupui.edu/research/EOF.

[28] "oregonFarming," 2020, https://www.oregon.gov/ODA/shared/Documents/Publications/Administration/ORAgFactsFigures.pdf.

[29] https://industrial-iot.com/2016/10/iot-oil-gas-real-power-can-provide/.

[30] "East texas oil field," 2020, https://en.wikipedia.org/wiki/East_Texas_Oil_Field.

[31] "WirelessHART System Engineering Guide," 2019, http://www2.emersonprocess.com/siteadmincenter/PM%20Central%20Web%20Documents/EMR_WirelessHART_SysEngGuide.pdf.

[32] "Emerson targets large scale deployment," 2018, http://www.automationworld.com/networking-amp-connectivity/wireless-shines-emerson-global-user-exchange.

[33] "SNOW Base station," 2019, https://github.com/snowlab12/gr-snow.

[34] "The future of agriculture? smart farming," https://www.forbes.com/sites/federicoguerrini/2015/02/18/the-future-of-agriculture-smart-farming/.

[35] "Farming industry must embrace the internet of things vision," http://www.beechamresearch.com/news.aspx?id=1086.

[36] M. H. Almarshadi, S. M. Ismail *et al.*, "Effects of precision irrigation on productivity and water use efficiency of alfalfa under different irrigation methods in arid climates," *Journal of Applied Sciences Research*, vol. 7, no. 3, pp. 299–308, 2011.

[37] H.-J. Kim, K. A. Sudduth, and J. W. Hummel, "Soil macronutrient sensing for precision agriculture," *Journal of Environmental Monitoring*, vol. 11, no. 10, pp. 1810–1824, 2009.

[38] N. D. Mueller, J. S. Gerber, M. Johnston, D. K. Ray, N. Ramankutty, and J. A. Foley, "Closing yield gaps through nutrient and water management," *Nature*, vol. 490, no. 7419, pp. 254–257, 2012.

[39] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "Farmbeats: An iot platform for data-driven agriculture," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI) 17)*, 2017, pp. 515–529.

[40] "FarmBeats: IoT for agriculture," 2018, https://www.microsoft.com/en-us/research/project/farmbeats-iot-agriculture/.

[41] "Monsanto," 2020, https://www.rcrwireless.com/20151111/internet-of-things/agricultural-internet-of-things-promises-to-reshape-farming-tag15.

[42] M. Handy, M. Haase, and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *4th IEEE International Workshop on Mobile and Wireless Communications Network*. IEEE, 2002, pp. 368–372.

[43] J. Henry, "802.11s mesh networking," 2011, https://www.cwnp.com/uploads/802-11s_mesh_networking_v1-0.pdf.

[44] 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.

[45] "NS3," 2019, https://www.nsnam.org.

[46] 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.

[47] "Industrial iot trends: Wsn, lpwan & cloud platforms," 2017, https://www.automation.com/pdf_articles/ON_world/InTechMagazineInsert_ONWorldFinal.pdf.

[48] 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.

[49] https://www.aliexpress.com/item/4001345191255.html?src=google&albch=shopping&acnt=494-037-6276&isdl=y&slnk=&plac=&mtctp=&albbt=Google_7_shopping&aff_platform=google&aff_short_key=UneMJZVf&&albagn=888888&albcp=9594035441&albag=102695258807&trgt=296904913880&crea=en4001345191255&netw=u&device=c&albpg=296904913880&albpd=en4001345191255&gclid=Cj0KCQjwxNT8BRD9ARIsAJ8S5xbA4Z8-uiZ51Y5CVHGlfxHtIvLUZrt6CQ8vh4rCwdf9Soirbl-W0T4aAsGvEALw_wcB&gclsrc=aw.ds.

[50] "TinyOS community forum," 2017, http://www.tinyos.net/.

[51] R. Prasad, *OFDM for wireless communications systems*. Artech House, 2004.

[52] C.-T. Chen, *System and Signal Analysis*. Thomson, 1988.

[53] 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.

[54] 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.

[55] O. D. Incel, "A survey on multi-channel communication in wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 3081–3099, 2011.

[56] X. Chen and J. Huang, "Game theoretic analysis of distributed spectrum sharing with database," in *2012 IEEE 32nd International Conference on Distributed Computing Systems*. IEEE, 2012, pp. 255–264.

[57] 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.

[58] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “A survey on spectrum management in cognitive radio networks,” *IEEE Communications magazine*, vol. 46, no. 4, pp. 40–48, 2008.

[59] 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.

[60] A. Saifullah, C. Wu, P. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen, “Near optimal rate selection for wireless control systems,” in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2012, pp. 231–240.

[61] Y. Chen and M. Chen, “Extended duality for nonlinear programming,” *Comput. Optim. Appl.*, vol. 47, pp. 33–59, 2010.

[62] V. P. Modekurthy, A. Saifullah, and S. Madria, “Distributedhart: A distributed real-time scheduling system for wirelesshart networks,” in *25th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2019, pp. 216–227.

[63] 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.

[64] http://www.ettus.com/product/details/UB210-KIT.

[65] https://www.ti.com/product/CC1310.