

Improving the Capacity of a Mesh LoRa Network by Spreading-Factor-Based Network Clustering

GUIBING ZHU¹, CHUN-HAO LIAO¹, THEERAT SAKDEJAYONT¹, I-WEI LAI²,
YOSHIKAKI NARUSUE¹, (Member, IEEE), AND HIROYUKI MORIKAWA¹

¹School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan

²Department of Electrical Engineering, National Taiwan Normal University, Taipei 10610, Taiwan

Corresponding author: Chun-Hao Liao (liao@mlab.tu-tokyo.ac.jp)

This work was supported in part by the Cross-Ministerial Strategic Innovation Promotion Program, Infrastructure Maintenance, Renewal, and Management Technology, in part by the JSPS Grant-in-Aid for Scientific Research (A) under Grant JP16H02358, and in part by the Research Laboratories, NTT Docomo, Inc.

ABSTRACT LoRa is a low-power long-range IoT standard that uses the chirp spread spectrum technique, and we have strived to further extend its coverage by utilizing the direct device-to-device (D2D) links to construct a multi-hop relay network. In LoRa, the spreading factor (SF) is an important parameter, which not only provides great flexibility between the data rate and sensitivity but also presents a new dimension for multiple accesses. Our approach to improving the capacity of a multihop LoRa network is to attempt to off-load the data traffic into several subnets by utilizing this multiple-access dimension. Each subnet rooted at a sink node is allocated a specific SF on the basis of network clustering. This enables packet transmission in parallel with multiple SFs to become feasible. To allow such parallel transmissions, our considerations are: 1) ensuring the connectivity of all subnets; 2) off-loading the traffic according to the number of nodes, data rates, and network topologies of each subnet; and 3) shortening the airtime of each subnet by reducing the hop count. Toward these objectives, we present a tree-based SF clustering algorithm (TSCA) to conduct SF allocation in a multihop LoRa network. The TSCA focuses on balancing the airtime between the subnets while ensuring connectivity. Furthermore, we use simulations to show that our approach can significantly increase the network performance compared with other approaches. We additionally deploy a real-chip experiment to evaluate the feasibility of parallel transmission in practice.

INDEX TERMS Low-power wide area network (LPWAN), LoRa, spreading factor (SF) allocation algorithm, tree-based spreading factor clustering algorithm (TSCA), multi-hop network.

I. INTRODUCTION

The LoRa standard, which is a Low-Power Wide-Area Network (LPWAN) specification [1]–[5], has garnered wide interest in the current Internet of Things (IoT) era. Recently, the mesh LoRa network has become a promising technology in both academic studies and industrial developments [6], [7]. In a mesh LoRa network, the nodes directly perform Device-to-Device (D2D) communications to relay the packets to the destination. Adopting a mesh topology not only allows us to extend the coverage by multiple relays, but more importantly provides the possibility to enhance the network efficiency [6].

However, researchers [8]–[11] also suggested that a performance enhancement such as this would require a proper

allocation algorithm regarding the most important parameter in LoRa namely the *spreading factor* of the chirp spreading spectrum (CSS). In this modulation, the spreading factor (SF) determines the slope of linear chirps and LoRa signals with different SFs are almost orthogonal to each other [12]–[15]. Therefore, the SF not only serves the purpose of throughput-range trade-off (similar to the MCS index in LTE or WiFi), but also provides an additional dimension for multiple access protocols such as CDMA. Specifically, the network utility can be significantly enhanced by allocating different SFs to the nodes and by transmitting packets in parallel.

In contrast to a single-hop network, SF allocation in a multi-hop LoRa network not only affects the network connectivity but also influences the flexibility in terms of the coverage, throughput, and latency. Specifically, the capacity of each SF in a single-hop network is determined only by the

The associate editor coordinating the review of this manuscript and approving it for publication was Dapeng Wu.

number of nodes and its data rate [16]. However, in a multi-hop LoRa network, the throughput of each subnet depends on the particular topology and this dependency has not yet been considered in previous research. Moreover, adopting parallel transmission by using SF allocation in a mesh LoRa is similar to using multiple channels in a multi-hop single-root data collection network. Wu *et al.* [17] have already proven multi-channel assignment as an NP-complete problem by reducing the k-coloring problem to assign a node to different channels. They presented a Tree-based Multi-channel Protocol (TMCP) in which a Breadth-First Searching (BFS) algorithm was adopted to construct a fat tree rooted at the sink node and then assigned the channel from the top layer to the bottom layer according to the topology. Similarly, Liew *et al.* [18] employed the first available channel algorithm that adopts BFS to assign a channel to a node after the first available channel is determined. However, these approaches ignored the balance between different subnets. In terms of channels, the coverage range they achieved was consistent, whereas the coverage was attenuated by using a smaller SF with a higher data rate.

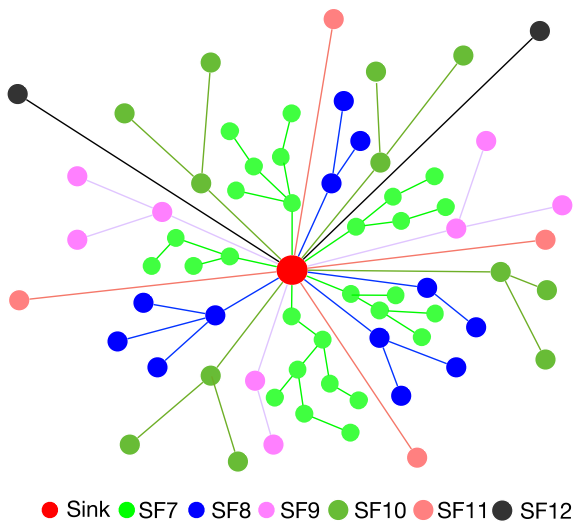


FIGURE 1. Conceptual topology of SF allocation with multiple subnets. Nodes marked with the same color use one specific SF to transmit packets. In each subnet, the connectivity of each of the nodes is well confirmed and enables the node to communicate with the sink node using the shortest path and the number of nodes is well balanced according to the capacity of each subnet.

The approach we followed to realize parallel transmissions in a multi-hop LoRa network, was to develop a conceptual structure of SF allocation, shown in Fig. 1, where all nodes in the network need to be allocated to several subnets and each subnet uses a separated SF. In addition, it was necessary to take the following considerations into account: 1) The connectivity of each subnet would have to be guaranteed after SF allocation. 2) The traffic load of each subnet would have to remain balanced because of the asymmetry of SFs (the data rate of SF7 is 18.6 times the data rate of SF12). 3) The hop count of the multi-hop network would have to be minimized to decrease the airtime of each transmission.

In this work, we proposed a tree-based SF clustering algorithm (TSCA) for realizing parallel transmission with a balanced SF allocation in a multi-hop LoRa network. Specifically, we consider the original network where all nodes keep using the fastest data rate (SF7) in the initial state. Moreover, given a well-connected tree using SF7, which is the shortest path from the relay nodes to the root node, the algorithm removes a certain number of nodes to generate spanning sub-trees using larger SFs. The TSCA addresses the aforementioned three considerations as follows:

1) As compared to a single-hop topology for which the connectivity check is unnecessary, the connectivity is confirmed when removing the nodes from the sub-tree that uses SF7 and inserting them into a higher SF node by node.

2) To balance the traffic loads between all sub-trees, we employ a constraint rule with not only the data rate, but also the tree height to limit the number of nodes in each sub-tree. The tree height is also considered when we first estimate the capacity of each SF by using the maximum hop-count on each sub-tree. Furthermore, the TSCA is designed such that the algorithm updates the constraint rule iteratively based on the real hop count of the current topology.

3) To reduce the hop count of each sub-tree, we attempt to remove the nodes farthest away from the root and using SF7, by inserting them into a sub-tree with a larger SF and with lower height. Specifically, we use the Bottom-up Breadth-First-Search (BBFS) algorithm to determine the order in which extraction occurs and the Top-down Breadth-First-Search (TBFS) algorithm for the insertion. Moreover, the balance between the sub-trees is maintained by inserting the removed nodes into a sub-tree that will have Minimal Air Time (MAT) after the insertion, instead of inserting the nodes into the Largest First-available SF (LFS) or the Smallest First-available SF (SFS).

The remainder of paper is organized as follows. In Section II, we present related work focused on the SF allocation problem in LoRa. We discuss the design concept of SF allocation in a mesh LoRa network in Section III. Section IV presents details of the proposed TSCA. We use simulations to evaluate the performance of TSCA in Section V. Section IV describes the real deployment we conducted to assess the feasibility of parallel transmission. We conclude the paper and discuss our future work in Section VII.

II. RELATED WORK

LoRa, a proprietary wireless communication standard promoted by the LoRa alliance, enables long-range communications. Even though the typical topology in LoRa is a single-hop network named LoRaWAN [19], [20], the SF allocation is an important issue because it improves the network efficiency in both single-hop and multi-hop LoRa networks.

A. SF ALLOCATION IN A SINGLE-HOP NETWORK

LoRaWAN, which is a single-hop network, implements an ALOHA or a slotted ALOHA mechanism on the Medium Access Control (MAC) layer with the physical design of

LoRa technology [21]. LoRaWAN ensures connectivity by standardizing the Adaptive Data Rate (ADR) mechanism to allow the node to step down its data rate. However, the ADR, which is based on the number of received acknowledgement (ACK) messages from gateways, is a basic method. These methods are inaccurate for assessing the highly-varying wireless environment, and render data transmission inefficient [22].

Subsequent research [23]–[26] considered an SF distribution scheme based only on the distance from the node to gateways. Adelantado *et al.* [23] showed that an excessive number of nodes (28 % of the network) should use the largest SF (SF12) to ensure the coverage of urban cells. This approach only considered the path loss and ignored the airtime when using SF12. Reynnders *et al.* [25] provided SF distribution scheme to balance the packet error rate [24] and lightweight scheduling to group the nodes into different power level and selected SFs to improve the reliability and scalability of the LoRaWAN network. Slabicki *et al.* [26] showed that a network-aware approach can further improve the delivery ratio of dense networks by using global knowledge of the node locations. However, the proposed algorithms based on ADR are not considered in the same way as for parallel transmission. Cuomo *et al.* [16] extended the work on parallel transmission in a single-hop LoRa network. Specifically, these authors proposed to use the airtime to balance the nodes of each group with a specific SF and to attempt to use a high data rate to offload the traffic to less congested larger SFs.

However, these strategies cannot be easily applied to a multi-hop LoRa network because of their lack of consideration in regards to multi-hop relays. In contrast to a single-hop network in which the airtime of different groups is only decided by the number of nodes and their data rates, the airtime in a multi-hop network is also determined by the hop count of each subnet. Moreover, the connectivity between multiple relays is still not considered when conducting SF allocation in a single-hop LoRa network.

B. SF ALLOCATION IN A MULTI-HOP NETWORK

As compared to single-hop LoRa networks, multi-hop networks are more flexible to extend the coverage and more efficient to improve the data transmission without increasing the number of gateways. A practical strategy that transforms the topology from a star to a mesh network when the coverage range exceeds 3.2 km was proposed [6]. Moreover, it was shown [27] that constructing a mesh LoRa network is a good solution to solve the coverage problem in extensively shadowed urban areas. However, few reports that discuss the SF allocation in a mesh LoRa network have been published.

On the other hand, similar to SF allocation in a mesh LoRa network, the adoption of parallel transmission by using multiple channels has already been implemented in conventional multi-hop single-root data collection networks. In data collection networks, the proposed protocols [17], [18] usually construct a static channel assignment approach to maintain the simplicity of channel coordination. Other researchers [17]

proposed TMCP to assign different channels to disjoint trees and operate parallel transmissions among sub-trees for data collection. However, the paper does not discuss the balance between different sub-trees. A more recent proposal [18] involved a multi-channel multi-path data collection protocol based on Basketball Net Topology (BNT), which maintains not only a tree-based topology but also the connectivities between peer nodes located at the same height in the tree. This protocol enables child nodes to rejoin the network, even when their parent node disappeared from the original tree structure, by using peer links to communicate with other nodes. However, the connectivities of peers extend the hop counts to the sink node, which inversely increases the airtime of the entire network.

The use of LoRa enables the coverage range to be extended when a lower data rate with a larger SF is chosen. As compared to multi-channel assignment algorithms, we needed to consider an approach that would decrease the hop count of each sub-tree using a different SF while ensuring that the airtime between different sub-trees remains balanced.

III. DESIGN CONCEPT OF SF ALLOCATION IN MULTI-HOP LoRa NETWORKS

In this section, we introduce the design concept of SF allocation in a mesh LoRa network. We describe the overall considerations in section III-A and discuss each of these considerations in the following subsections.

A. OVERALL CONSIDERATIONS

To achieve capacity improvement by parallel transmission, we aim to separate one full-connected network into several disjointed subnets with different SFs. Specifically, each node is allocated with a single SF and the nodes using the same SF forms a multi-hop sub-network that operates independently with the others. Under such a scenario, three aspects need to take into consideration to maximize the capacity of the whole network: 1) The traffic needs to be offloaded to each subnet in a balanced way to prevent any subnet becoming the throughput bottleneck; 2) Each subnet needs to be well-connected without any unreachable nodes; 3) With a proper SF allocation, the diameter (network height) of each subnet can also potentially be reduced so that the capacity is further improved. In the following subsections, we discuss how the proposed TSFA achieve the aforementioned three aspects.

B. BALANCING THE CAPACITY WITH EACH SF

To protect a subnet from being allocated with too many nodes, we determine the number of nodes in each subnet according to their capacity. The number of nodes in each subnet M_i can be given by

$$M_i = M \times \frac{C_i}{\sum_{j=7}^{12} C_j}, \quad i \in \{7 \sim 12\}, \quad (1)$$

where M is the total number of nodes in the network and C_i is the capacity of a subnet with SF i .

One straightforward way to estimate the capacity is to use the throughput rate of each SF, namely

$$C'_i = R_i, \quad (2)$$

where R_i is the data rate of the subnet with SF i . This estimation has been adopted to the single-hop scenario, such as LoRaWAN [16]. Therefore, in this work, we refer to this method as the Single-hop Estimation (SHE).

Different from adopting SHE in a multi-hop LoRa network, the capacity of each subnet can maintain a more acceptable balance when taking the height of the subnet into consideration. Accordingly, we propose to use the Maximum-hop Estimation (MHE) to estimate the capacity of each subnet where C_i can be written as

$$C''_i = \frac{R_i}{H_{\max,i}}, \quad (3)$$

and $H_{\max,i}$ is the maximum hop count of the subnet with SF i and determines the worst case of data transmission. Note that, Eq. (2) is a subset of Eq. (3) where the value of $H_{\max,i}$ is always one. Contrary to SHE, MHE estimates each $H_{\max,i}$ by allocating a specific SF i to all nodes, respectively. Moreover, the algorithm conducts the SF allocation iteratively where MHE will be executed in the first iterative cycle. In subsequent cycles, the algorithm calculates the resultant $H_{\max,i}$ based on the previous cycle to further balancing each C_i .

Using Eq. (3), we adopt a constraint rule to limit each M_i and rewrite Eq. (1) as

$$M_i = M \times \frac{R_i}{H_{\max,i} \times \sum_{j=7}^{12} \frac{R_j}{H_{\max,j}}}, \quad i \in \{7 \sim 12\}. \quad (4)$$

To simplify the analysis, we assume a common data collection scenario where each node generates an equal amount of data periodically and those data need to be transmitted to the sink node within a given period of time. When inserting too many nodes into a subnet that is over its capacity, the time of finishing the transmission in this subnet will be longer than the others and becomes the bottleneck of data collection.

Therefore, in this work, the proposed algorithm aims to minimize the maximum airtime of each subnet. After SF allocation, the transmission time of each node is a product of the airtime of the data transmitted per bit and the actual $H_{\max,i}$. Then, the definition of the airtime of each subnet can be written as

$$T_i = M_i \times \frac{H_{\max,i}}{R_i}, \quad i \in \{7 \sim 12\}, \quad (5)$$

where T_i is the airtime of the subnet using SF i . Accordingly, to evaluate the performance of conducting parallel transmission, the airtime of the entire network is defined as

$$T = \max_{i \in \{7 \sim 12\}} \left(\frac{1}{T_i} \right), \quad (6)$$

where T is the maximum airtime between the subnets. Furthermore, during the node insertion and extraction process, we also calculate the dynamic T_i and assign the

TABLE 1. List of notations for TSCA.

Notation	Definition
i	Spreading Factor, $i \in \{7 \sim 12\}$
n_j	the node of index j , $1 < j < M$
M	the total number of nodes in the network
T_i	the airtime of the subnet with SF i
$\Theta_{h,i}$	set of nodes of height h of the sub-tree with SF i
$\Pi_i(n_j)$	set of parents of node n_j with SF i
$\Phi_i(n_j)$	set of children of node n_j with SF i

extracted node to the emptiest subnet with the MAT to maintain the balance among the subnets.

C. ENSURING CONNECTIVITY

Another important aspect of SF allocation is to ensure the connectivity of each subnet. For every process that determines a node needs to be extracted from the original network and then inserted into a subnet with a larger SF, the algorithm guarantees connectivity. More specifically, we assume that, initially, the network is well connected with SF7. The node extraction process checks whether all child nodes of the current determining node have at least one remaining parent node after the current one is extracted from the original network. After the node is earmarked for extraction from the subnet with SF7, the insertion process would also need to ensure that the extracted node has an available parent node in a spanning subnet with a larger SF with which to communicate, otherwise, the node would remain with SF7 to ensure connectivity.

D. REDUCING THE HEIGHT OF SUBNETS

In order to minimize T , the algorithm is necessary to reduce each $H_{\max,i}$ which is strongly affected by the order in which nodes are extracted and inserted during SF allocation. An effective order of extraction allows $H_{\max,7}$ to be reduced whereas an effective order of insertion ensures that the height of the other subnets remains small.

When reducing $H_{\max,7}$, it is beneficial to extract more nodes from the bottom layer of the tree. Further, it is preferable to insert the extracted node by starting from the upper layer to prevent the height from becoming excessive. Consequently, we use the BBFS algorithm to determine the order of extraction from SF7 and the conventional TBFS algorithm to ensure the extracted node is inserted into a sub-tree with a larger SF, which is the shortest path to the root node. Moreover, the subnets identified for accepting the extracted nodes need to be prioritized.

IV. TREE-BASED SF CLUSTERING ALGORITHM (TSCA)

The flowchart of the proposed TSCA is shown in Fig. 2 and the corresponding pseudocode is presented in Algorithm 1. The notations are listed in Table 1. As shown in the figure, the main parts are as follows:

A. INITIALIZATION

After the algorithm builds a well-connected network with SF7 in the initial phase, the root node retains the one-hop

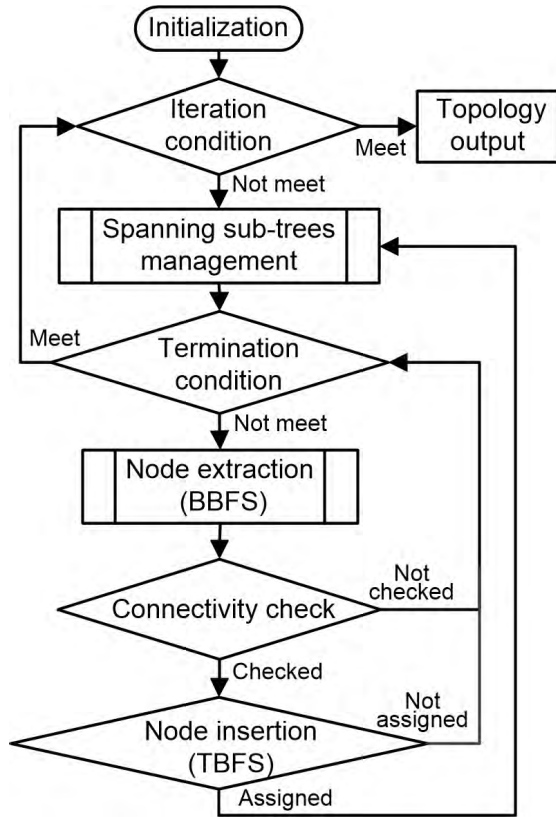


FIGURE 2. Flowchart showing the scheme of the TSCA. After initialization, the algorithm changes the iterative conditions according to the capacity of each SF in every iterative cycle. The termination condition manages the node extraction, connectivity check, and node insertion processes in each iterative step. The algorithm outputs the topology of each subnet after all nodes are allocated to a specific SF.

knowledge for all relay nodes and records their hop counts of all the existing shortest paths from each relay node to the root node. We define the topology of a sub-tree with SF i as a set $\Theta_{h,i}$, where h is the hop count to the sink node. For every node n_j , the parent set and child set are $\Pi_s(n_j)$ and $\Phi_s(n_j)$, respectively.

The simulation assumes the root node is able to communicate with its first-hop relay nodes by using all SFs in parallel. Moreover, according to the coverage of each SF, the algorithm builds the topologies under all SFs where all nodes use a specific SF from 7 to 12 in one subnet.

B. ITERATIVE CONDITION

The TSCA is designed to estimate the capacity of each subnet by using the constraint rule (Eq. 1). In step 3 of Algorithm 1, the iterative condition is set to limit the number of nodes in each sub-tree according to the estimation of MHE, which is the maximum hop count of each subnet we calculated from the initialization phase in the first iterative cycle. After the first cycle, the algorithm calculates the current topology information and then renews the iterative condition according to the actual hop count of each sub-tree.

Algorithm 1 TSCA: Tree-Based SF Clustering Algorithm

Input: $\Theta_{h,7}; \Theta_{h,s \neq 7} \begin{cases} \{root\} & h = 1 \\ \emptyset & h > 1 \end{cases}; \Pi_i(n_j)$ and $\Phi_i(n_j)$ set of every node n_j .

Output: $\Theta_{h,s}$

```

1: START
2: // Constraint rule changed with iterations
3: while Iteration condition not met do
4:   while Termination condition not met do
5:      $\{n_j\} \leftarrow \text{nodeExtract}(\Theta_{h,7})$ 
6:     for  $j = 1 : \text{end do}$ 
7:        $Checked \leftarrow \text{conCheck}(\Theta_{h,7}, n_j)$ 
8:       if  $Checked$  then
9:          $Agn \leftarrow \text{nodeAssign}(\Theta_{h,8 \sim 12}, n_j)$ 
10:        if  $Assigned$  then
11:          //Spanning Sub-tree Management
12:           $\Theta_{h,i} \leftarrow \text{treeMgn}(\Theta_{h,i}, n_j, Agn)$ 
13:          break;
14:   return  $\Theta_{h,i}$ 
15: function BBFS: nodeExtract( $\Theta_{h,7}$ )
16:   // BBFS (default), bottom-up
17:   for  $h = H_{max,7} : 2$  do
18:     for  $\forall n_j \in \Theta_{h,7}$  do
19:        $\{n_j\} \leftarrow n_j$ 
20:   return  $\{n_j\}$ 
21: function conCheck( $\Theta_{h,7}, n_j$ )
22:    $Checked = 0$ 
23:   if  $(\forall n'_x \in \Phi_7(n_j), n_j \in \Pi_7(n'_x))$  then
24:      $Checked = 1$ 
25:   return  $Checked$ 
26: function nodeAssign( $\Theta_{h,8 \sim 12}, n_j$ )
27:    $Agn = 0$ 
28:   for  $i = 8 : 12$  do
29:     // TBFS, from top to bottom
30:     for  $h = 1 : H_{max,i}$  do
31:       if  $\exists n'_x \in \Theta_{h,s}, n'_x \in \Pi_i(n_j)$  then
32:          $\{i\} \leftarrow i$ 
33:       // MAT (default), subnet  $i$  with the minimal airtime,
34:        $i \leftarrow \min(T_i), i \in \{i\}$ 
35:   return  $Agn = i$ 
36: function treeMgn( $\Theta_{h,s}, n_j, Agn$ )
37:   if  $Assigned$  then
38:      $\Theta_{h,Agn} \leftarrow n_j$ 
39:      $\Theta_{h,7} = \Theta_{h,7} - n_j$ 
40:   return  $\Theta_{h,s}$ 
41: END
  
```

Every iterative cycle is terminated when the number of nodes in each subnet meets the iterative condition or when this cycle has no more nodes that can actually be extracted from the original network after the algorithm searched all nodes with SF7. If no additional nodes are extracted in the

next iterative cycle as compared to the previous one, the simulation is terminated as the final iterative cycle.

C. TERMINATION CONDITION

The termination condition is determined according to every iterative condition. In step 4 of Algorithm 1, the termination condition takes charge of every node assignment process in an iterative step, which includes a successful assignment, the connectivity check being false, and the insertion being false. When the constraint is met in this iterative cycle or the algorithm searches all nodes in the original network with SF7, the termination condition terminates the iteration and outputs the topology information to renew the next iterative condition.

D. SPANNING SUB-TREES MANAGEMENT

We show the function for the management of spanning sub-trees in step 36 of Algorithm 1. This function generates five empty sub-trees rooted at the sink node in the initial phase, where $\Theta_{h,i}$ is \emptyset when $i \neq 7$ and $h > 1$. For every successful node assignment, the management process deletes node n_j from sub-tree $\Theta_{h,7}$ and spans the sub-tree Θ_{h,Ag_n} into which node n_j is inserted.

E. NODE EXTRACTION PROCESS

In order to determine the order in which to extract the nodes from the original subnet, the BBFS first searches the nodes in the bottom layer where h is equal to $H_{max,i}$ and then proceeds to a higher layer as shown in step 15. After an identified node is assigned to a larger SF successfully, BBFS runs the search from the bottom layer again in an attempt to extract additional nodes from the lower layers. After the algorithm searches all nodes with SF7, the algorithm terminates the current iteration.

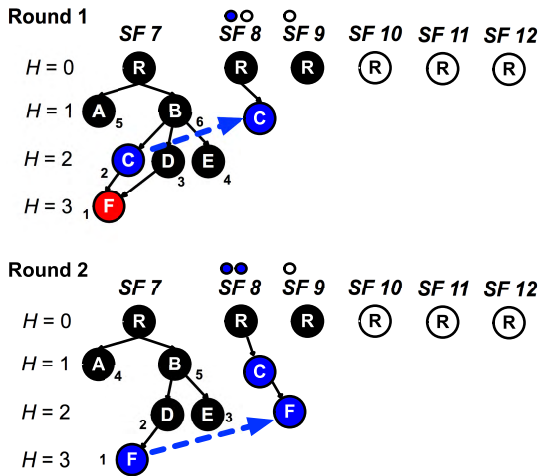


FIGURE 3. Conceptual procedure of the TSCA with six child nodes in total. The capacity of SF8 and SF9 is two nodes and one node, respectively. The small numbers beneath the nodes in the sub-tree using SF7 indicate the order of extraction. The red relay node is an example of a failed extraction because of the connectivity check whereas the blue node indicates successful SF allocation to join a subnet with a larger SF.

An example of SF allocation is shown in Fig. 3, where we illustrate the conceptual process of node extraction that

consists of two rounds in one iterative cycle. In the first round, BBFS determines the extraction order from 1 to 6 and checks the connectivity from node F . The first extracted node is node C , which can communicate with the root node by using SF8. Then, the algorithm starts to search the extraction sequence from the bottom layer again in round 2. Because node C is assigned to SF8, node F can communicate with the root node in two hops by using SF8. The algorithm continues to search for a possible node that can be assigned to SF9 until the end of this round.

On the other hand, we compare BBFS with the Breadth-First-Searching algorithm (BFS) [17], [18] and the Depth-First-Searching algorithm (DFS) which show different strategies to perform the node extraction process. As a reference, we present the pseudocode of the BFS and DFS functions in Algorithm 2. In the case of BFS, the algorithm starts to traverse the tree at the sink node and checks all of the neighboring nodes at the current height. After all nodes are explored, the algorithm moves on to the nodes at the next height. Using Fig. 3 as an example, the visiting order is A, B, C, D, E, F where node A and node E can be extracted to subnet SF8. However, because of the connectivity problem, the extracting node F cannot be assigned to subnet SF9 even though SF9 is not filled to capacity. As opposed to BFS, DFS traverses the tree from the sink node and explores each branch of the tree as far as possible by recursively calling the pre-order function. In Fig. 3, the visiting order is A, B, C, F, D, E when using DFS and the extracted nodes are A, C, F, E in the first round. Node A and node C are inserted into subnet SF8 and node F still faces a connectivity problem with regard to being inserted into SF9.

F. NODE INSERTION PROCESS

The node insertion function, in step 26 of Algorithm 1, adopts TBFS to search all sub-trees for which SF is larger than 7 and the search proceeds from the top layer to the bottom layer until the first possible parent node n'_x is found for node n_i . The algorithm records all possible SFs to insert the node. The algorithm maintains the height of the spanning sub-trees at the lowest possible level by selecting a sub-tree into which to insert the nodes based on the computation of MAT.

Moreover, we compared the insertion priorities with the first-available channel algorithm [18], in which we present the LFS and SFS as different strategies of the node insertion process. After a node was extracted from subnet SF7, LFS chooses the first-available largest SF and SFS chooses that of the smallest SF to assign the node.

G. CONNECTIVITY CHECK

For any child node n_x of extracted node n_j , the connectivity check confirms whether more than one parent node with SF 7 exists. Then the algorithm sets the flag *Checked* to 1 if all child nodes have at least one other parent node with which to communicate.

Lastly, we present an example of SF allocation with 200 nodes in Fig. 4. As shown in the figure, the nodes located

Algorithm 2 Overriding the Node Extraction Function

Input: $\Theta_{h,7}$;
Output: $\{n_j\}$

```

1: START
2: // Overriding the function to 1)BBFS, 2)BFS, 3) DFS
3:  $\{n_j\} \leftarrow \text{nodeExtract}(\Theta_{h,7})$ 
4: END
5: function BFS: nodeExtract( $\Theta_{h,7}$ )
6:   // BFS: Breadth-first-searching
7:   for  $h = 2 : H_{max,7}$  do
8:     for  $\forall n_j \in \Theta_{h,7}$  do
9:        $n_j.\text{visited} = \text{ture}$ 
10:       $\{n_j\} \leftarrow n_j$ 
11:   return  $\{n_j\}$ 
12: function DFS: nodeExtract( $\Theta_{h,7}$ )
13:   // DFS: Depth-first-searching
14:   init(){
15:     for  $h = 2 : H_{max,7}$  do
16:       for  $\forall n_j \in \Theta_{h,7}$  do
17:          $n_j.\text{visited} = \text{false}$ 
18:         for  $\forall n_j \in \Theta_{h,7}$  do
19:           DFS( $\Theta_{h,7}, n_j$ ) }
20:   DFS() {
21:      $n_j.\text{visited} = \text{ture}$ 
22:     for  $\forall n_x \in \Phi_s(n_i)$  do
23:       if  $n_x.\text{visited} == \text{false}$  then
24:         DFS( $\Theta_{h,7}, n_x$ ) }
25:   return  $\{n_j\} \leftarrow \{n_j.\text{visited}\}$ 

```

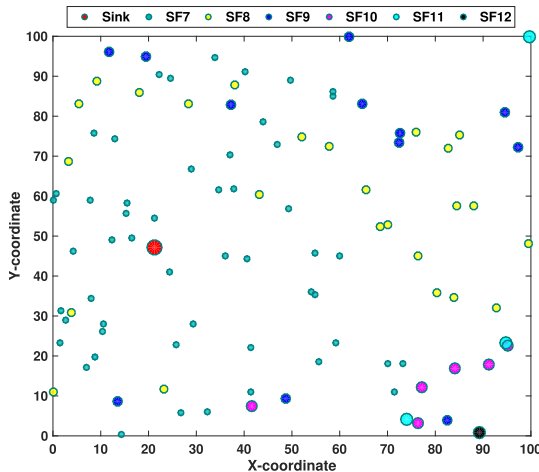


FIGURE 4. Example of spatial distribution with 200 nodes in 2-D space. The sink node is marked with red color. Nodes marked with the same color are allocated to one subnet with a specific SF. First, the nodes are allocated to different SFs in a balanced manner - about half of the nodes are distributed to SF7, half of the remainder to SF8, and so forth. Second, the nodes near the sink node are allocated to smaller SFs, whereas those far away from the sink are assigned to larger SFs. This allocation helps to reduce the hop count of packet relays.

close to the sink node are assigned a smaller SF and the nodes farther from the sink node are assigned a larger SF. The nodes in the subnets with different SFs are well balanced

with a declining trend of the number of nodes using a larger SF. In addition, the nodes using a larger SF stay further from the sink node, which maintains the hop count of each subnet small.

V. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed TSCA by simulations. We describe the simulation setup in section V-A. Before presenting the detail, we first show that the TSCA can achieve a more optimal balance as compared to other approaches in section V-B. In section V-C, we further discuss the performance of different capacity estimation methods, tree-searching algorithms for the node extraction process, and assignment order for the node insertion process. We also discuss the effect of the number of iterative cycles.

A. SIMULATION SETUP

In the simulation, we assume the nodes are uniformly and randomly distributed on a rectangular space related to the length (L units) and width (W units) of its sides. We also assume that one unit is equal to the coverage of SF7 and increasing SF by 1 extends the coverage range by a factor of $\sqrt{2}$ based on the free space model. We assume that connectivity between two nodes exists only if one node is within the coverage of another node and all nodes are well connected by using SF7.

We simulate two scenarios. The first involves a square space (L = 5, W = 5) as the normal scenario. We additionally consider another scenario with long length and narrow width (L = 25, W = 1), which creates a linear topology with a large hop count. The number of nodes is increased from 20 to 200. For each number of nodes and each topology, we run the simulations up to 10,000 times. According to previous research [2], the coverage of SF7 is roughly 1 km in an urban area and the simulated scenario covers areas of approximately 25 km².

B. EVALUATION OF THE DEGREE OF BALANCE BETWEEN EACH SF

In this subsection, we describe the evaluation of the balance of each SF by using TSCA. In Fig. 5, we compare the different strategies of SF allocation by simulating each T_i with SF i . The rate at which the airtime decreases is shown by normalizing each T_i to that of the network using only SF7. In an effective balanced scenario between the subnets, the airtime of all subnets should be as close as possible such that the airtime of the network becomes the smallest by adopting parallel transmission. The number of nodes in the network can be assumed to be 100 in the following discussions unless otherwise stated.

In each subplot in the figure, the results show that MHE outperforms SHE, with MHE capable of offloading more traffic into the larger SFs and balancing the airtime of each subnet. As compared to each column of the plot, the proposed BBFS, which is designed to extract more nodes from the bottom side of the original network, decreases the airtime of subnet SF7 most effectively compared to BFS and DFS.

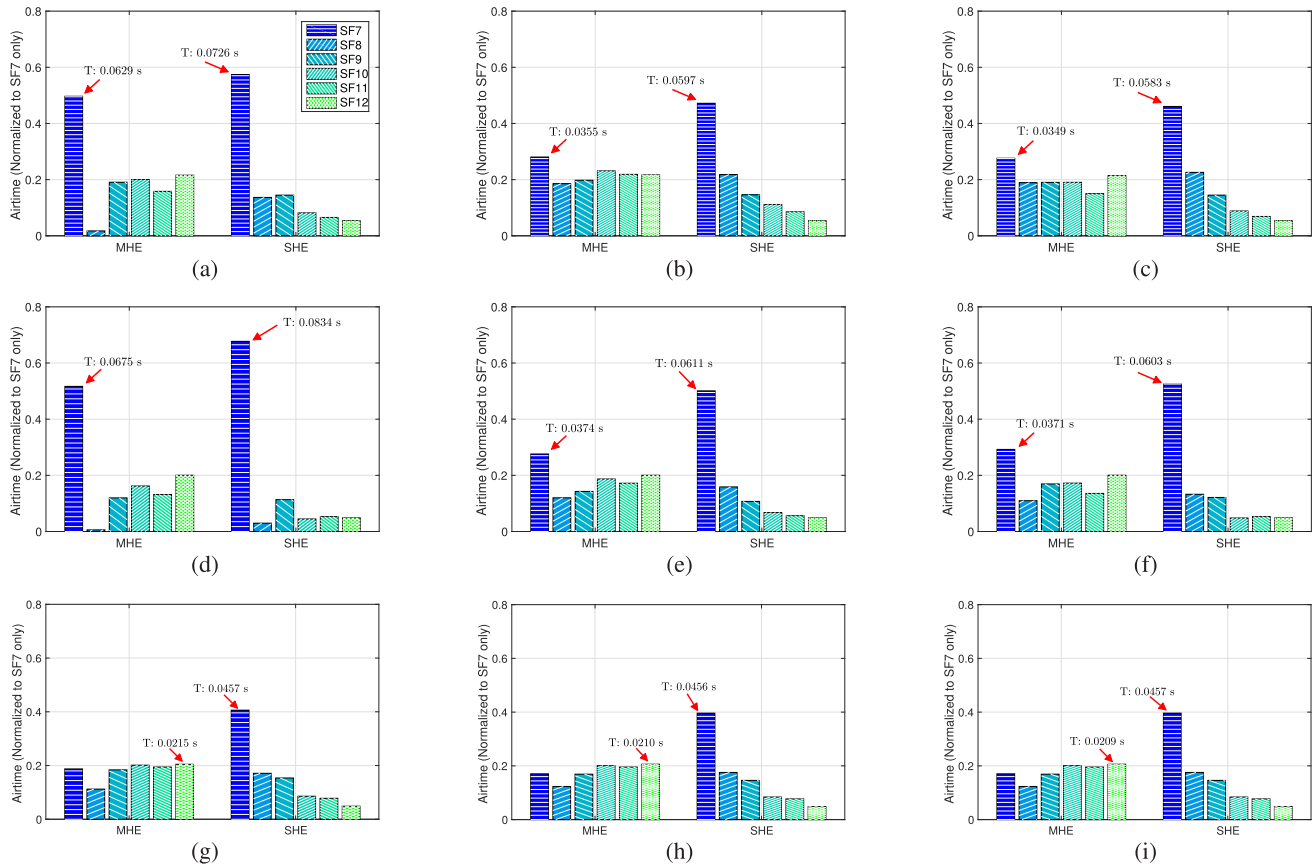


FIGURE 5. Balance of the capacity of each subnet with the different comparisons. In each subplot, the x-axis shows the two capacity estimation methods, MHE and SHE, respectively, and the y-axis shows the airtime of a subnet with a value normalized to that of all nodes using only SF7. For each set of bars, the subnets with different SFs are distinguished by different colors and patching objects according to the legend. Moreover, in each subplot, we identify the maximum airtime of subnets with an arrow pointing to the particular subnet. (a) DFS-LFS. (b) DFS-SFS. (c) DFS-MAT. (d) BFS-LFS. (e) BFS-SFS. (f) BFS-MAT. (g) BBFS-LFS. (h) BBFS-SFS. (i) BBFS-MAT.

As compared to each row of the plot, the proposed MAT minimizes the airtime of the network as opposed to when using LFS and SFS. Fig. 5 (a) suggests the allocation loses the balance of subnet SF8 where the airtime is largely less than that of the other subnets. Furthermore, in Fig 5 (d), the large amount of airtime for subnet SF7 indicates the failure to insert nodes into subnets with larger SFs. As shown in Fig. 5 (i), the airtime of the network reduces to roughly 20% of that of using SF7 only by running TSCA.

C. EVALUATION OF THE EFFECT OF EACH STRATEGY OF TSCA

In this subsection, we further discuss the effect of the SF allocation based on each strategy of TSCA. To make the comparison fair, the proposed MHE, BBFS, and MAT are set as default when one of the three have been chosen for comparison with other approaches and the other two are not mentioned.

1) CAPACITY ESTIMATION METHODS

The effect of capacity estimation methods is investigated by evaluating the extent to which MHE can outperform SHE.

Fig. 6 shows the estimated number and the actual allocated number of nodes in each subnet by using two methods, respectively. The results in Fig. 6 (a) show that the allocated number of nodes can achieve the estimated capacity of each subnet by using both MHE and SHE in the area defined as a square. MHE outperforms SHE by extracting more nodes from SF7 and allows the insertion of more nodes into subnets with larger SFs in a balanced way.

On the other hand, in the long and narrow rectangular area, the results in Fig. 6 (b) suggest that MHE and SHE did not achieve the estimated capacity of the subnets with smaller SFs (SF8, SF9). Specifically, the limited coverage prevents the insertion of nodes that are difficult to insert into subnets because of their connectivity. Because of this, MHE can allocate additional capacity to the larger SFs and still outperforms SHE with more nodes inserted to the subnets.

2) ORDER OF NODE EXTRACTION

To demonstrate the importance of determining the node extraction order, we compare the performance of different tree searching algorithms including BBFS, BFS, and DFS. Our evaluation shows that BBFS can effectively lower the

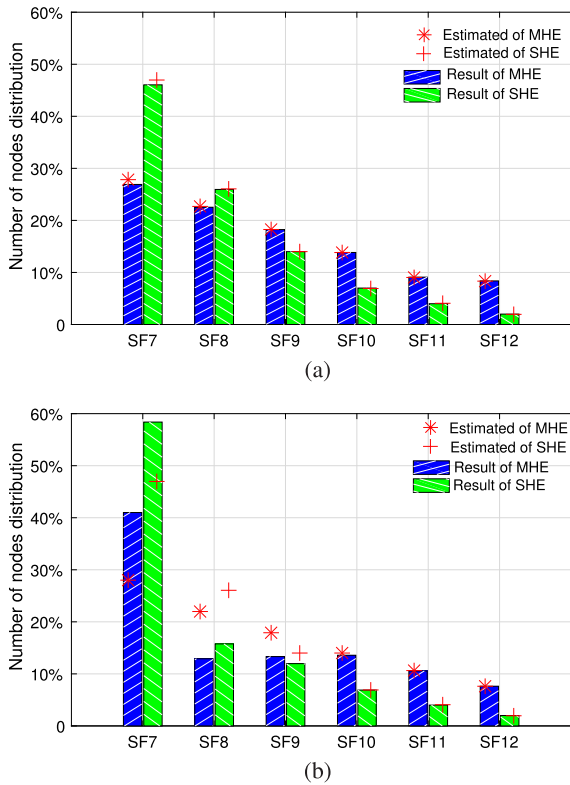


FIGURE 6. Number of nodes distributed for each subnet: (a) scenario in which nodes are distributed in a $100 \times 100 \text{ m}^2$ square space; (b) scenario in which nodes are distributed in a $20 \times 500 \text{ m}^2$ rectangular space. The x-axis shows the subnets of each SF and the y-axis the percentage of nodes distributed. The asterisk and plus signs represent the estimated number of nodes of each subnet by using MHE and SHE, respectively. The blue and green bars are the actual number of nodes assigned to each subnet after the first iterative cycle, respectively.

tree height (maximum hop count) of subnet SF7 as compared to the other two methods. The results in Fig. 7 show the probability of the hop count decreasing and the averaged decrease in the hop counts by adopting different tree searching algorithms. Fig. 7 (a) shows that, in the square space, BBFS outperforms BFS and DFS with the most simulation rounds that can decrease the number of hops of a subnet by a larger percentage. An average 84 % of simulation rounds cannot reduce the height of subnet when using BFS, and the maximum hop count is only decreased one hop with 50 % of simulation rounds when using DFS. The result shows that the use of BBFS can reduce the height of the original network by an average of 41 % in the square space.

Fig. 7 (b) shows, in the rectangular space, the searching algorithms undergo a performance degradation with no significant decrease in the hop counts. In this scenario, BFS and DFS can barely reduce the height. In this space, the performance of the proposed BBFS is weaker as compared to the square space and the result shows that the tree height can be reduced by an average of 12 % when the maximum hop count is significantly increased.

Further, we evaluate the airtime of the network by increasing the number of nodes in the square space. Fig. 8 shows

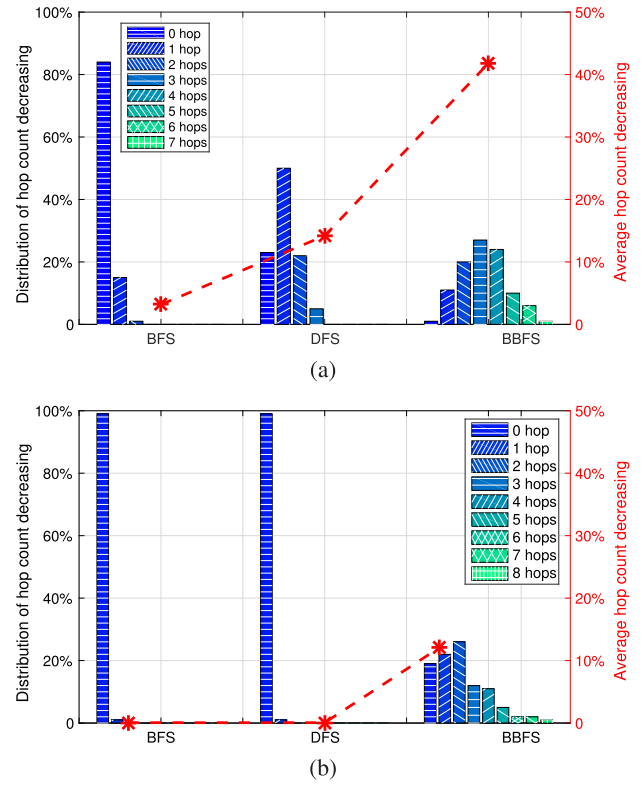


FIGURE 7. Decrease in the hop count of the original network: (a) square space distribution and (b) rectangular space distribution. The x-axis represents the different searching algorithms. The y-axis on the left shows the percentage reduction of the distribution of the hop count where the legend distinguishes the number whereby the hop count was decreased for each bar. The y-axis on the right indicates the ratio of the average decreased hop count as compared to the original maximum hop count of the network. The square and rectangular spaces achieve an average maximum hop count of 7 and 24 hops by using SF7, respectively.

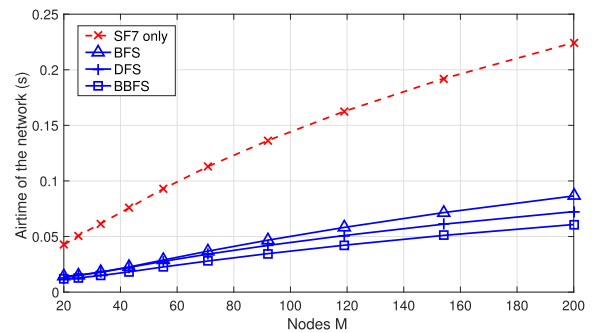


FIGURE 8. Decrease in the airtime of the network. Nodes are distributed in the same square space. The x-axis shows the number of nodes in the network. The y-axis on the left represents the airtime of the network. The red dashed line shows the airtime in the absence of parallel transmission. The blue lines with different markers show different extraction orders by using different searching algorithms. Note that MHE and MAT were adopted and the results are based on the first iterative cycle.

parallel transmission to be an effective approach as compared to the network without parallel transmission. The result demonstrates that BBFS is more effective to decrease the airtime of the network as compared to the other two algorithms.

3) ORDER OF NODE INSERTION

We also evaluate the performance of using different strategies to determine the node insertion order including MAT, LFS, and SFS. In Fig. 5 (a-f), the results show, when adopting BFS and DFS, that MAT performs the best by achieving the lowest airtime of the network. Moreover, Fig. 5 (g-i) shows, when BBFS is adopted, the performance of different insertion strategies is similar. The airtime of each subnet is more balanced and the difference in the airtime required by the network is small when MHE and BBFS are adopted.

4) ITERATIVE CYCLES

We evaluated the effect of balancing the airtime of the network by running additional iterative cycles. The ratio of the airtime of the network between parallel transmission and the network using only SF7 is shown. The comparison is between conducting SF allocation without iteration and that of conducting SF allocation iteratively by using MHE and SHE.

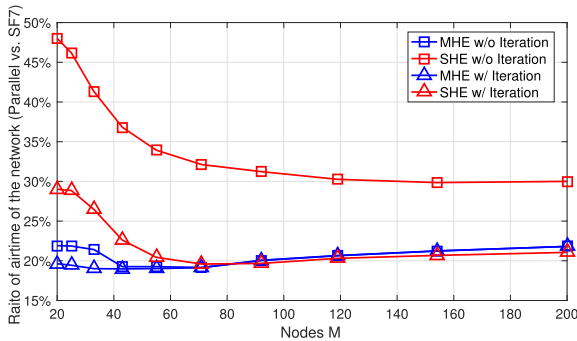


FIGURE 9. Effect of using iteration. The red and blue lines represent the results of SHE and MHE, respectively. The triangular and square markers represent the results obtained without iteration and those obtained iteratively, respectively. The x-axis is the number of nodes and the y-axis shows the ratio of airtime of the network as compared to all nodes using SF7. The extraction and insertion were performed with BBFS and MAT.

The results in Fig 9 show that running the allocation iteratively can further decrease the airtime of the network. When using SHE, the airtime of the network decreased largely by running iterative cycles. When using MHE, the performance of running more iterative cycles is only improved with a low-density scenario where the number of nodes is small.

TABLE 2. Necessary number of iterative cycles.

	Low density (M=50)		High density (M=100)	
	SHE	MHE	SHE	MHE
Probability of 1 cycle	0	41.1%	0	8.1%
Probability of 2 cycles	95.6%	57.6%	22.4%	71.2%
Probability of 3 cycles	4.4%	1.3%	70.4%	19.5%
Probability of 4 cycles	0	0	7.2%	1.2%
Average iterative cycles	2.04	1.60	2.85	2.14

On the other hand, in order to evaluate the rate of convergence of using MHE and SHE, we show the necessary number of iterative cycles. As shown in Table 2, the number

of nodes is 50 in a low-density scenario and 100 in a high-density scenario. The results show that, in both low-density and high-density scenarios, SHE needs a larger number of iterative cycles than MHE. Moreover, the number of iterative cycles increases with an increase in the number of nodes.

To analyze the overall performance of the iteration, we discuss the performance loss when using SHE as compared to MHE. The one reason is the difficulty of balancing the airtime among the subnets. In the low-density scenario, a slight difference in the number of nodes in the subnets is not expected to change the calculated capacity of the largest SFs (SF11, SF12). However, the rounding errors of the number of nodes in the subnets vary the airtime of the network considerably whereas MHE allows more nodes to be inserted into the largest SFs.

Moreover, although BBFS decreased the height of subnet SF7 substantially, the nodes extracted from the bottom of the subnet have a lower possibility to be inserted into the subnet with smaller SFs (SF8, SF9) when considering the coverage. This complicates the achievement of the calculated capacity of the subnets (SF8, SF9) as shown in Fig. 6. On the other hand, in the high-density scenario, SHE outperforms MHE slightly. This is because the increased connectivity allows the use of SHE to more effectively balance the subnets by running additional iterative cycles.

VI. EXPERIMENTAL RESULTS

This section describes the feasibility of parallel transmission in a mesh LoRa network by using multiple SFs. The efficiency of the proposed TSCA was further evaluated by additionally conducting real-chip experiments in a practical environment. Specifically, we used 36 LoRa RF modules fitted with a Semtech SX1276 and STM32L0 microprocessor to cover two main buildings on the campus. Because the LoRa end-point devices cannot normally communicate with multiple SFs simultaneously, six nodes were assigned to act as the sink node with one specific SF and remain close to each other to emulate the parallel transceiving function of LoRa gateway. We show the distribution of all the nodes deployed in the experiment in Fig. 10.

TABLE 3. Parameter space of the experiment.

Parameter	Values
Spreading factor	7 to 12
Tx power	13 dBm
Center frequency	920.6 MHz
Bandwidth	500 kHz
Code rate	4/ 5
Packet length	8 bytes

The preparation for SF allocation entailed collecting the connectivity information between any pair of nodes. The specific setup and parameter space are listed in Table 3. Fig. 11 shows the topology of all nodes using one SF where each link exists with a Packet Reception Rate (PRR) of more than 90 % and each path is the shortest path to the

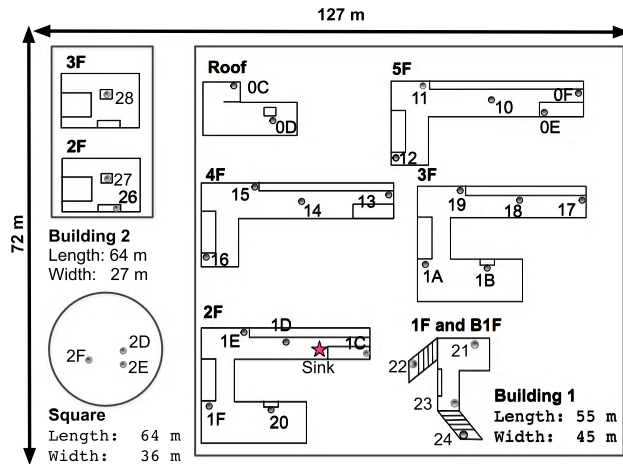


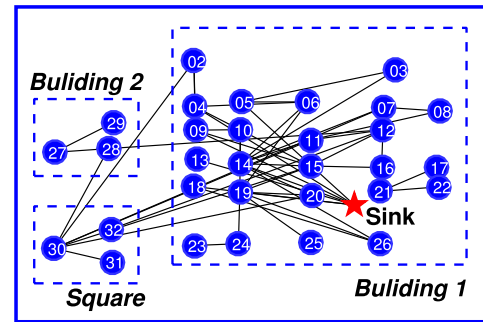
FIGURE 10. Node deployment in the experiment. Most of the nodes are located in Building 1 and three nodes are deployed in Building 2 with three other nodes located in the square. Sufficient indoor coverage is ensured by thoroughly shielding the nodes within a lidless iron box or a cupboard.

TABLE 4. Experimental results of SF allocation.

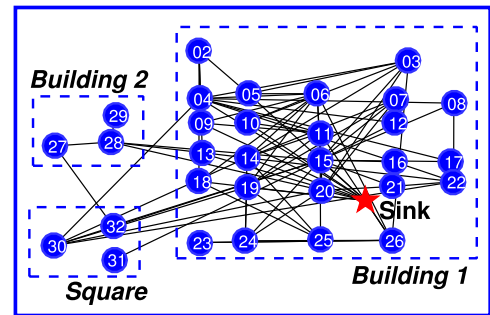
		BBFS	DFS	BFS	Only SF7
SF7	M_7	13	14	17	30
	$H_{max,7}$	6	6	6	7
SF8	M_8	7	4	4	0
	$H_{max,8}$	3	2	2	0
SF9	M_9	4	5	3	0
	$H_{max,9}$	2	3	3	0
SF10	M_{10}	3	3	3	0
	$H_{max,10}$	2	2	2	0
SF11	M_{11}	2	2	2	0
	$H_{max,11}$	2	2	2	0
SF12	M_{12}	1	2	1	0
	$H_{max,12}$	1	2	1	0
Maximum airtime (T)		0.0143	0.0154	0.0187	0.0384
Airtime reduction ratio		62.8 %	60.0 %	51.3 %	N/A
Throughput enhancement		2.69	2.50	2.05	N/A

sink node. Fig. 11 (a) and (b) present the cases for SF7 and SF12, respectively. Accordingly, this enabled us to gather the one-hop knowledge of each node for the connectivity check. The proposed algorithm is executed off-line which needs to collect the connectivity information from a static topology. A variation of topology (e.g., mobile nodes) may lead to the necessity of regular update on the connectivity information which consumes the time and energy in practice. The development of a method to conduct fast topology collection in practice is beyond the scope of this paper.

The experiment was designed to evaluate whether parallel transmissions degenerate the overall performance of the network by analyzing the airtime and the PRR. Based on different SF allocation methods, we compare different searching algorithms after using MHE and MAT and execute the algorithm iteratively. Based on the results in Table 4, we can compute that BBFS decreases the network airtime by 62.8 % as compared to the network using only SF7. On the other hand, BFS and DFS reduce the airtime by 51.3 % and 60 %, respectively.



(a)



(b)

FIGURE 11. Topology of the network when using (a) SF7 and (b) SF12, respectively. As compared to SF7, even though the maximum hop count decreased when using SF12, the connectivity of the nodes increases where a node is able to communicate with additional nodes within one hop.

respectively. The result shows that, by conducting parallel transmission in a mesh LoRa network, the achievable throughput of the network is improved $1/(1 - 0.628) = 2.69$ times as compared to all nodes using single SF7.

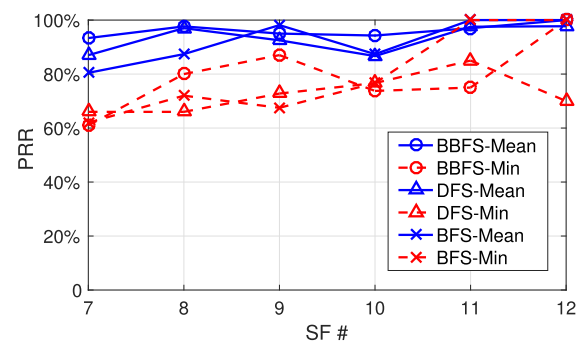


FIGURE 12. Experimental results for the comparison of different SF allocation methods. The x-axis represents the respective subnets using a specific SF and the y-axis shows the PRR calculated for each subnet. The blue solid lines show the mean PRR and the red dashed lines show the minimal PRR.

In addition, we collected the PRR of the subnets for each SF by adopting a concurrent transmission protocol for data collection in LoRa [14]. Fig. 12 shows the PRR of the TSCA as compared with other approaches. The results show that the mean PRR of each subnetwork is approximately 90 % when the TSCA is used. The use of multiple SFs to achieve parallel

transmission did not result in extensive interference nor did it cause the performance of the entire network to degenerate. Moreover, the largest SF (SF11, SF12) shows the strongest interference resilience whereas SF7 is the most affected by collision.

VII. CONCLUSIONS AND FUTURE WORK

The results presented in this paper showed that a multi-hop LoRa network offers the possibility of parallel transmission with multiple SFs. Furthermore, we demonstrated that the construction of a more efficient multi-hop LoRa network needs to take SF allocation into account as a matter of priority. Our approach was to develop a tree-based SF clustering algorithm (TSCA) to assign the nodes to several subnets. The algorithm was designed to ensure the connectivity of each subnet, after which the airtime of subnets is balanced by conducting SF capacity estimation with the consideration of the number of nodes, data rates, and the hop count of subnets. The proposed TSCA also uses a bottom-up breadth-first searching algorithm (BBFS) to determine the order in which to extract the nodes from the original network, which uses SF7. We additionally aimed to insert the extracted nodes such that the airtime of the particular subnet was minimized to ensure that the airtime between subtrees remained balanced. The evaluation confirmed a large performance improvement as compared to the SF allocation in a single-hop LoRaWAN network and the tree-based multi-channel assignment approaches in WSN.

Our future plans include working on further improving the efficiency of parallel transmission in mesh LoRa networks. Also, we are studying a fast connectivity prediction approach which could use the fastest data rate to predict the connectivity of all SFs.

REFERENCES

- [1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 855–873, 2nd Quart., 2017.
- [2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 60–67, Oct. 2016.
- [3] M. Saari, A. M. B. Baharudin, P. Sillberg, S. Hyrynsalmi, and W. Yan, "LoRa—A survey of recent research trends," in *Proc. MIPRO CTS*, May 2018, pp. 1035–1040.
- [4] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.
- [5] D. Ismail, M. Rahman, and A. Saifullah, "Low-power wide-area networks: Opportunities, challenges, and directions," in *Proc. Workshops ICDCN*, Jan. 2018, pp. 8:1–8:6.
- [6] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda, "Evaluating LoRa energy efficiency for adaptive networks: From star to mesh topologies," in *Proc. IEEE WiMob*, Oct. 2017, pp. 1–8.
- [7] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-hop LoRa networks enabled by concurrent transmission," *IEEE Access*, vol. 5, pp. 21430–21446, 2017.
- [8] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of LoRaWAN channel access," in *Proc. EnT*, 2016, pp. 10–14.
- [9] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Apr. 2017.
- [10] E. R. Lin, "LoRa protocol. Evaluations, limitations practical test," M.S. thesis, Univ. Politècnica Catalunya, Barcelona, Spain, May 2016.
- [11] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology," in *Proc. ITST*, 2015, pp. 55–59.
- [12] *LoRa Modulation Basics*, SEMTECH, Camarillo, CA, USA, Apr. 2016.
- [13] L. Vangelista, "Frequency shift chirp modulation: The LoRa modulation," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017.
- [14] G. Zhu, C.-H. Liao, M. Suzuki, Y. Narusue, and H. Morikawa, "Evaluation of LoRa receiver performance under co-technology interference," in *Proc. IEEE CCNC*, Jan. 2018, pp. 1–7.
- [15] K. Mikhaylov, J. Petäjäjärvi, and J. Janhunen, "On LoRaWAN scalability: Empirical evaluation of susceptibility to inter-network interference," in *Proc. EuCNC*, 2017, pp. 1–6.
- [16] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations," in *Proc. ACM/IEEE WiMob*, Oct. 2017, pp. 1–8.
- [17] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1193–1201.
- [18] S.-Y. Liew, C.-K. Tan, M.-L. Gan, and H. G. Goh, "A fast, adaptive, and energy-efficient data collection protocol in multi-channel-multi-path wireless sensor networks," *IEEE Comput. Intell. Mag.*, vol. 13, no. 1, pp. 30–40, Feb. 2018.
- [19] N. Sornin and A. Yegin, "LoRaWAN specification v1.1," LoRa Alliance, Beaverton, OR, USA, Tech. Rep., 2017.
- [20] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bi-directional traffic do more harm than good in LoRaWAN based LPWA networks?" in *Proc. GLOBECOM*, Dec. 2017, pp. 1–6.
- [21] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan, and M. Gidlund, "Scalability analysis of a LoRa network under imperfect orthogonality," *IEEE Trans. Ind. Informat.*, to be published.
- [22] D.-Y. Kim, S. Kim, H. Hassan, and J. H. Park, "Adaptive data rate control in low power wide area networks for long range IoT services," *J. Comput. Sci.*, vol. 22, pp. 171–178, Sep. 2017.
- [23] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Sep. 2017.
- [24] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *Proc. IEEE ICC*, May 2017, pp. 1–6.
- [25] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of LoRaWANs through lightweight scheduling," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1830–1842, Jun. 2018.
- [26] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks for dense IoT deployments," in *Proc. IEEE/IFIP NOMS*, Apr. 2018, pp. 1–9.
- [27] K.-H. Ke, Q.-W. Liang, G.-J. Zeng, J.-H. Lin, and H.-C. Lee, "A LoRa wireless mesh networking module for campus-scale monitoring: Demo abstract," in *Proc. ACM/IEEE IPSN*, Apr. 2017, pp. 259–260.



GUIBING ZHU received the B.S. and M.E. degrees in automation engineering from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree with The University of Tokyo. His research interests include wireless sensor networks, embedded systems, and machine-to-machine systems. He received the Best Paper Award at the IEEE Consumer Communications and Networking Conference, in 2018.



CHUN-HAO LIAO received the B.S. degree in electrical engineering and the M.S. degree in electronics engineering from National Taiwan University, Taipei, Taiwan, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from The University of Tokyo, Tokyo, Japan, in 2017, where he is currently a Project Researcher with the School of Engineering. From 2008 to 2009, he was a Research Assistant with the Institute for Integrated Signal Processing Systems,

RWTH Aachen University, Germany. From 2009 to 2014, he was a Communication System Designing Engineer with MediaTek Inc. His research interests include baseband signal processing of wireless communication systems, VLSI system design, and network protocols. He received the Best Paper Award at the IEEE Consumer Communications and Networking Conference, in 2018.



THEERAT SAKDEJAYONT received the B.E. degree in electronics and communication engineering from the Sirindhorn International Institute of Technology, Pathum Thani, Thailand, in 2011, and the M.E. and Ph.D. degrees from the Graduate School of Engineering, The University of Tokyo, Tokyo, Japan, in 2014 and 2017, respectively, where he is currently a Project Researcher with the Department of Electrical Engineering and Information System, Graduate School of Engineering.

His research interests include wireless sensor networks, digital baseband wireless communication, and wireless communications for control systems.



I-WEI LAI received the Ph.D. degree in electrical engineering from the Graduate Institute of Electronics Engineering (GIEE), National Taiwan University, in 2011. He was with RWTH Aachen University, Germany, and with Academia Sinica, Taiwan. Since 2017, he has been with the Department of Electrical Engineering, National Taiwan Normal University, as an Assistant Professor. His research interests include signal processing, optimization, and performance analysis for wireless

communication networks. He is a member of the Phi Tau Phi Scholastic Honor Society. He was a recipient of the NTU-GIEE Best Thesis Award, in 2011, for his Ph.D. thesis. He was recognized as the Exemplary Reviewer of the IEEE TRANSACTIONS ON COMMUNICATIONS, in 2014.



YOSHIAKI NARUSUE (M'13) received the B.E., M.E., and Ph.D. degrees from the Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan, in 2012, 2014, and 2017 respectively, where he is currently a Research Associate with the Department of Electrical Engineering and Information Systems. His research interests include wireless power transfer, wireless communication, and the Internet of Things. He is a member of the IEICE and IPSJ. He received the

Second-Best Student Paper Award at the IEEE Radio and Wireless Symposium, in 2013, the Hiroshi Harashima Academic Encouragement Award, in 2013, the TECO Green Tech Contest Bronze Award, in 2016, and the Best Paper Award at the IEEE Consumer Communications and Networking Conference, in 2018.



HIROYUKI MORIKAWA received the B.E., M.E., and Dr.Eng. degrees in electrical engineering from The University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively, where he is currently a Full Professor with the School of Engineering. From 2002 to 2006, he was a Group Leader with the NICT Mobile Networking Group. His research interests include ubiquitous networks, sensor networks, big data/IoT/M2M, wireless communications, and network services. He is a Fellow of

IEICE. He has received more than 50 awards, including the IEICE Best Paper Award, in 2002, 2004, and 2010, the IPSJ Best Paper Award, in 2006, the JSCICR Best Paper Award, in 2015, the Info-Communications Promotion Month Council President Prize, in 2008, the NTT DoCoMo Mobile Science Award, in 2009, the Rinzauro Shida Award, in 2010, the Radio Day Ministerial Commendation, in 2014, and the IEEE CCNC Best Paper Award, in 2018. He served as a Technical Program Committee Chair for many IEEE/ACM conferences and workshops, the Vice President of IEICE, the OECD Committee on Digital Economy Policy Vice Chair, and the Director of New Generation M2M Consortium. He serves on numerous telecommunications advisory committees and frequently serves as a Consultant to government and companies.

...