Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5017319, 16 pages
https://doi.org/10.1155/2018/5017319

Research Article

Critical-Path Aware Scheduling for Latency Efficient Broadcast in Duty-Cycled Wireless Sensor Networks

Duc-Tai Le,[1] Giyeol Im,[1] Thang Le Duc,[1] Vyacheslav V. Zalyubovskiy,[2] Dongsoo S. Kim,[3] and Hyunseung Choo[1]

[1] *Sungkyunkwan University, Jangangu, Seobu-ro 2066, Suwon 440-746, Republic of Korea*
[2] *Sobolev Institute of Mathematics, 4 Acad. Koptyug avenue, 630090 Novosibirsk, Russia*
[3] *Indiana University-Purdue University Indianapolis, 420 University Blvd., Indianapolis, IN 46202, USA*

Correspondence should be addressed to Hyunseung Choo; choo@skku.edu

Received 27 June 2018; Revised 18 September 2018; Accepted 9 October 2018; Published 22 October 2018

Academic Editor: Alessandro Bazzi

Minimum latency scheduling has arisen as one of the most crucial problems for broadcasting in duty-cycled Wireless Sensor Networks (WSNs). Typical solutions for the broadcast scheduling iteratively search for nodes able to transmit a message simultaneously. Other nodes are prevented from transmissions to ensure that there is no collision in a network. Such collision-preventions result in extra delays for a broadcast and may increase overall latency if the delays occur along critical paths of the network. To facilitate the broadcast latency minimization, we propose a novel approach, critical-path aware scheduling (CAS), which schedules transmissions with a preference of nodes in critical paths of a duty-cycled WSN. This paper presents two schemes employing CAS which produce collision-free and collision-tolerant broadcast schedules, respectively. The collision-free CAS scheme guarantees an approximation ratio of $(\Delta - 1)T$ in terms of latency, where Δ denotes the maximum node degree in a network. By allowing collision at noncritical nodes, the collision-tolerant CAS scheme reduces up to 10.2 percent broadcast latency compared with the collision-free ones while requiring additional transmissions for the noncritical nodes experiencing collisions. Simulation results show that broadcast latencies of the two proposed schemes are significantly shorter than those of the existing methods.

1. Introduction

Wireless Sensor Networks (WSNs) have become one of the most important technologies for the 21st century [1, 2] and have attracted a lot of attention from industrial and research perspectives [3–11]. A wide range of services in WSNs rely strongly on broadcasting, such as information dissemination, route discovery, and code update [12]. Implementing an effective network-wide broadcast scheduling is critical to improve the performance of WSNs. Like other wireless communications, broadcasting can suffer from collision when a node hears more than one message simultaneously from transmissions of its neighbors. If the collision occurs, the node is not able to receive any of these messages. The problem of Minimum Latency Broadcast Scheduling (MLBS) aims to schedule a broadcast such that a message will be disseminated from a source node to all other nodes in a network with a minimum period of time. The problem has been proven to be NP-hard and widely studied in conventional WSNs, where all nodes are active all the time [13–15].

Recently, duty-cycle protocols let nodes in a network turns off their radios and go into a sleep state when they are idle to conserve energy and to extend network lifetime [16–18]. In a low-duty-cycled WSN, a sensor node sleeps most of the time periodically waking up within a short time slot for possible data communication by either transmitting or receiving a message. Such periodic sleeping leads to a notable increase of communication latency between sensor nodes while reducing their energy consumption. In particular, every sensor node has to wait until its receivers wake up before transmitting a message and may need to transmit the message more than once if its receivers wake up at different time slots. The MLBS problem in duty-cycled WSNs gets more complex and has been proved as a NP-hard problem [19].

Table 1: Summary of broadcasting algorithms in duty-cycled WSNs.

	Multichannel	Unreliable link	Clock drift	Collision
Enhanced LAyered Coloring [19]	No	No	No	Yes
Vector-Iteration Algorithm [20]	No	No	No	Yes
One-To-All Broadcast [21]	No	No	No	Yes
Latency-Aware Broadcast Scheduling [22]	No	No	No	Yes
Opportunistic flooding [23]	No	Yes	No	Yes
NAB [24]	Yes	No	No	Yes
Delay-Driven-Tree-based opportunistic flooding [25]	No	Yes	No	Yes
Load-Balanced Parents Assignment Algorithm [26]	No	No	No	No
MultiChannel Broadcast [28]	Yes	No	No	Yes
Degree-based Collision Tolerant Scheduling [29]	No	No	No	Yes
Minimum-Delay Energy-efficient flooding Tree [30]	No	Yes	No	Yes

Throughout the literature, the MLBS in duty-cycle (MLBSDC) problem has been richly explored [19–30]. The existing solutions typically aim to find a subset of nodes that can simultaneously transmit a message at a time slot so that the number of collision-free receptions in the time slot is maximized. However, delaying a transmission to prevent collision at a node in a network may increase overall broadcast latency if the node is in a critical path of the network. On minimizing broadcast latency, we propose a novel approach, critical-path aware scheduling (CAS) that offers an opportunity to reduce the latency by providing a preference for transmitting a message to nodes along critical paths of a duty-cycled WSN.

This paper presents two scheduling schemes employing CAS, Collision-Free-CAS (CF-CAS) and Collision-Tolerant-CAS (CT-CAS). Both schemes assign a numerical value for each node of a network to identify the critical paths during their scheduling phases. CF-CAS scheme does not allow any collision in a schedule to minimize the number of redundant transmissions in a broadcast. CT-CAS scheme further reduces broadcast latency by allowing collisions at noncritical nodes to speed up the broadcast process for critical nodes. To complete a broadcast schedule, it covers nodes experiencing collisions by additional transmissions. The increase of number of transmissions in CT-CAS reveals a trade-off between the redundancy and latency of a broadcast. Simulation results demonstrate the advantage of CAS in minimizing broadcast latency in duty-cycled WSNs. Broadcast latencies of both schemes are shorter than those of existing methods.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 includes a network model, problem formulation, and related terminology. In Section 4, we present the proposed schemes. Sections 5 and 6 show their performance analysis and evaluations, respectively. Finally, we conclude the paper and discuss our future work in Section 7.

2. Related Work

Research on the minimum-latency broadcast problem has increased over the past few decades. One of the earliest works, Gandhi et al. [13], proves the NP-hardness of the MLBS problem in the Unit Disk Graph (UDG) model, where all nodes have the same transmission range. Their proposed algorithm gives an approximation ratio in terms of latency of at least 400. Huang et al. [14] exploited the fact that 12 colors are sufficient to color all the nodes in any independent set of an UDG and thus the distance between nodes with same color is more than two hops. As a result, they improved the approximation ratio to 16. The algorithm in [15] recently reduced the ratio to 12 by allowing a node to transmit more than once to reduce broadcast latency.

However, the above-mentioned algorithms fail to capture the intermittently connected characteristic of duty-cycled networks. Several issues for broadcasting in duty-cycled networks have been explored, such as minimum latency [19–26, 28–30], minimum number of transmissions [31–34], and energy-saving [35, 36]. The MLBSDC problem rises as one of the most crucial problem for broadcasting in duty-cycled networks. Many research efforts have been made in literature to tackle the problem under different network settings, such as single channel [19–22, 26, 29], multichannel [24, 28], and unreliable links [23, 25, 30]. Table 1 summarizes the characteristics of the broadcasting algorithms in duty-cycled WSNs.

The paper revisits MLBSDC problem in a time-synchronized network, in which all nodes use the same channel and the communication links between them are reliable. The problem is proven as NP-hardness [19], and there is a handful of related work so far. Duan et al. [20] developed the Vector-Iteration Algorithm (VIA) which transforms a broadcast scheduling into a matrix multiplication. This work introduces several matrices and vectors to characterize the essentials of broadcast scheduling in duty-cycled WSNs. VIA has a high computational complexity for matrix multiplication. In order to reduce broadcast latency, the proposed scheme greedily searches for a set of forwarding nodes at each time slot such that as many nodes as possible can receive a broadcast message in the time slot. VIA produces a collision-free broadcast schedule with an approximation ratio of ΔT, where Δ is the maximum node degree in a network.

The One-To-All Broadcast (OTAB) scheme [21] utilizes a correlation function between network topology information

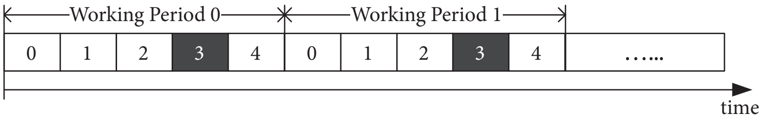


Figure 1: Duty-cycle schedule of a node with an active slot 3 and $T = 5$.

and the sleep schedule of each node to find the minimum latency from a source node to every node in a network. All nodes in the network are divided into layers, according to the minimum latency. The scheme applies a D2-coloring method on the MIS of each active slot and schedules transmissions layer-by-layer based on the assigned colors. Also, OTAB requires all forwarding nodes in a one-hop propagation to finish their transmissions before all neighbors in the next hop. As a result, it increases the delay to ensure collision-free transmissions from those one-hop neighbors and hence leads to a high broadcast latency. The scheme has an approximation ratio of $17T$, where T denotes the number of time slots in a working period.

Latency-Aware Broadcast Scheduling (LABS) [22] divides all nodes in a network into different layers according to their minimum latencies from the source node. It further reduces broadcast latency by employing an independent scheduling between consecutive layers of the network. The scheme schedules transmissions at several time slots in a single working period to maximize the number of receptions in the working period. It also explores geometric properties of the MIS to reduce the number of transmissions. A D2-coloring method is applied to prevent interference between transmissions for nodes within one layer. The broadcast schedule provided by LABS reduces up to 34 percent of latency while keeping the same approximation ratio as OTAB. The total number of transmissions of LABS is at most 10 times as large as the minimum number of transmissions.

Recently, Le et al. [29] proposed the Degree-Based Collision Tolerant Scheduling (DCTS) scheme. The scheme first categorizes internal and leaf nodes in a degree-based broadcast tree as primary and secondary nodes of a network, respectively. In order to minimize broadcast latency, it then speeds up the broadcast process for the primary nodes by selectively allowing collisions at secondary nodes of the network. DCTS accommodates such collisions by additional transmissions, thus ensuring the completion of a broadcast schedule. The scheme guarantees an approximation ratio of $(\Delta - 1)T$. Simulation results show that the scheme reduces to at least 94 percent of the broadcast latency compared with LABS, while slightly increasing the number of transmissions due to the additional transmissions.

On minimizing broadcast latency, we have discovered that scheduling a broadcast with a preference of nodes in critical paths of a duty-cycled WSN is beneficial to reduce latency. The intuition behind this is that broadcast latency may increase if a delay occurs along critical paths of the network. In [37], such critical paths are determined on an arbitrary *Shortest Path Tree (SPT)* constructed by a one-to-all shortest path algorithm, such as Dijkstra's algorithm [38]. Based on the critical-path awareness, we have sketched a collision-free schedule [37] and a collision-tolerant schedule [39] and presented preliminary simulation results to show the advantage of critical-path aware scheduling in reducing broadcast latency.

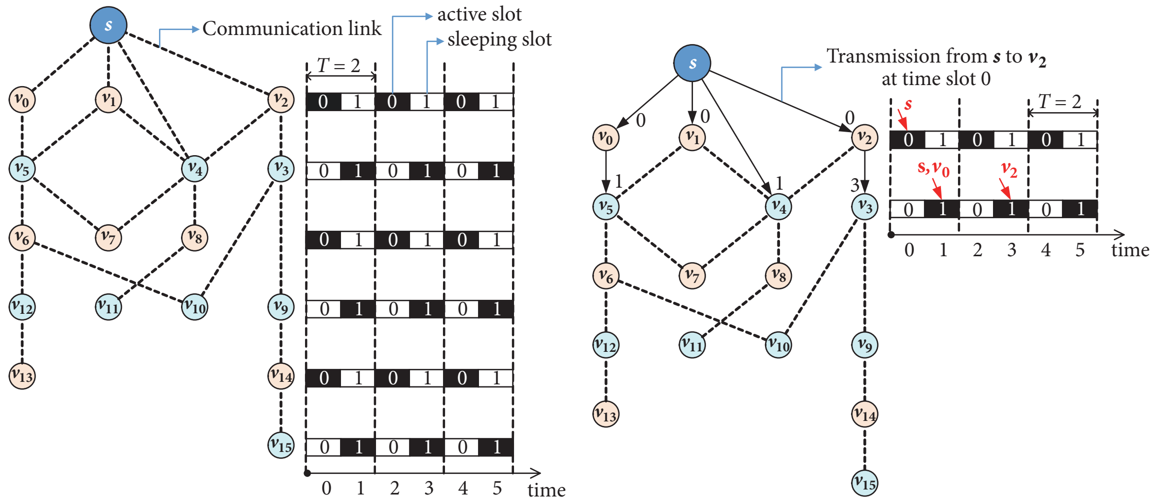
In this paper, we determine the critical paths on an SPT constructed with a preference of high-degree nodes to reduce the number of transmissions in a broadcast. In addition to a collision-free schedule, we develop a collision-tolerant schedule, similar to the one in [29], by referring to nodes in such critical paths as primary nodes. Moreover, both collision-free and collision-tolerant approaches adaptively select forwarding nodes in a broadcast, instead of selecting parents to forward a message to their children in a preconstructed broadcast tree as in [22, 29]. Such flexible selections also contribute to the reduction of broadcast latency in a critical-path aware schedule. Theoretical analysis and simulation results in this paper demonstrate a significant improvement of the proposed scheduling on broadcast latency.

3. Preliminary

3.1. Network Model and Assumptions. We consider a WSN of uniformly deployed sensor nodes in a square field as in [19–22, 29]. Each sensor node in the network is assigned a unique identifier. Two sensor nodes form a bidirectional communication link and become neighbors whenever they are within their transmission ranges of each other. The network topology is modeled as a graph, in which each vertex and each edge correspond to a node and a communication link between two nodes in the network, respectively. Such a graph is called a communication graph and assumed to be connected.

In duty-cycled environments, sensor nodes alternate between *active* and *sleep* state to conserve their energy. Time is divided into unit *time slots*. These discrete time slots are grouped into multiple *working periods*, with fixed length T. It is assumed that each time slot is long enough to accommodate the transmission of a message. All sensor nodes are time synchronized at the slot level using local time synchronization techniques, such as Reference-Broadcast Synchronization [40], Tiny-Sync [41], and Generalized ML-Like Estimator [42]. A sensor node u randomly selects one time slot in $\{0, 1, \dots, T-1\}$ as its active slot $a(u)$. It periodically wakes up at $a(u)$ for each working period and stays in the active state for the time slot. Figure 1 explicitly illustrates an example of the periodic sleeping schedule.

A sensor node can forward a broadcast message only after it receives the message. A sensor node can wake up at any time slot to transmit a message and can receive a message only at its active slot. Due to the properties of a wireless environment, whenever a node transmits a message, all its



(a) Communication graph along with active slots of nodes

(b) A two-hop broadcast schedule

Figure 2: Broadcasting in duty-cycled WSNs.

active neighbor nodes hear the message. If a node hears more than one message at a time slot, it cannot receive any of these messages due to a collision. We assume that every transmission occupies a unit time slot and no message suffers from bit error. Therefore, sensor node u successfully receives a message if only one neighbor of u transmits the message at time slot $a(u)$ of a working period.

3.2. Problem Statement. In a one-to-all broadcast, a message is disseminated from a source node to all other nodes in a network. The broadcast completes when every node in the network receives the message. To be specific, let $G = (V, E)$ denote the communication graph of a duty-cycled WSN, where V is the set of vertices, and E is the set of edges. Let s denote a predefined source node of the network. The source node starts a broadcast at time slot 0.

A *broadcast scheduling* assigns transmitting time slots for nodes in the network such that the broadcast can complete. As node $u \in V \setminus \{s\}$ can receive a message only at its active time slot, one of its neighbors must be assigned a transmitting time slot in a form of $iT + a(u)$ for a nonnegative integer i. The *broadcast latency* is determined by the maximum assigned transmitting time slot. The objective of the MLBSDC problem is to find broadcast schedule with a minimum broadcast latency. The MLBSDC problem is an NP-hard problem [19].

Figure 2 shows a simple duty-cycled WSN and a schedule for broadcasting a message from s to its two-hop neighbors in the network. In the schedule, nodes v_0, v_1, and v_2 receive a message from one transmission of s. After the transmission, nodes s and v_0 simultaneously transmit the message at time slot 1 to their corresponding neighbors v_4 and v_5, respectively. In contrast, s and v_2 cannot transmit the message at the same time to prevent a collision at their common neighbor node v_4. As a result, a transmission from v_2 is delayed to the next working period. The two-hop broadcast completes after 4 time slots.

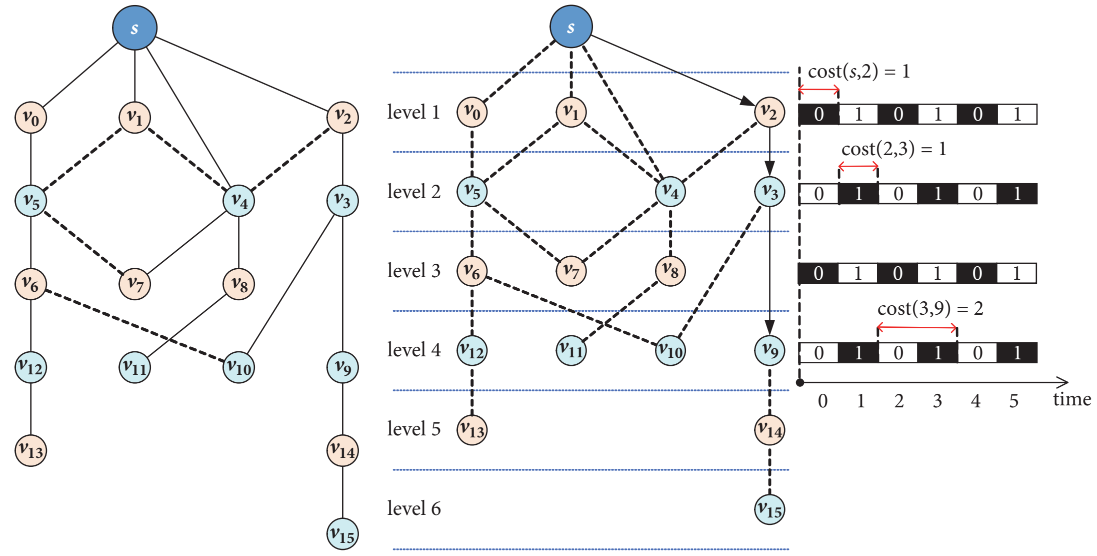
3.3. Related Terminology. For each pair of neighboring nodes in a communication graph $G = (V, E)$, we define two numerical values, called *costs*, corresponding to two asymmetrical directions of the edge between the nodes. Each cost value is the number of time slots for which a transmission on a direction of the edge is delayed due to the sleeping period of its receiver. For simplicity, the source node s is assumed to have a message in advance, and its active slot is defined as $T - 1$. The cost of an edge $(u, v) \in E$ with a direction from u to v can be determined as follows:

$$cost\,(u, v) = \begin{cases} a(v) - a(u), & \text{if } a(v) > a(u) \\ a(v) - a(u) + T, & \text{otherwise} \end{cases} \quad (1)$$

The minimum of accumulated costs on the paths from source node s to node $u \in V$ is referred to as *level* of u, denoted as $lv(u)$. The level of u corresponds to the minimum possible latency for u to receive a broadcast message generated by s; i.e. the smaller the level of a node is, the sooner the node may receive a broadcast message. The maximum level of nodes in a network is the lower bound of broadcast latency [19]. As the source node s has generated the message in advance, its level is 0. Levels of other nodes in the network can be obtained by constructing an SPT rooted at s. The SPT construction utilizes cost values as the weight of the edges. An example of SPT construction and level distribution for the network in Figure 2 is shown in Figure 3. It is worth noting that every node with a level k $(k > 0)$ has the same active slot $(k - 1) \bmod T$.

4. Proposed Schemes

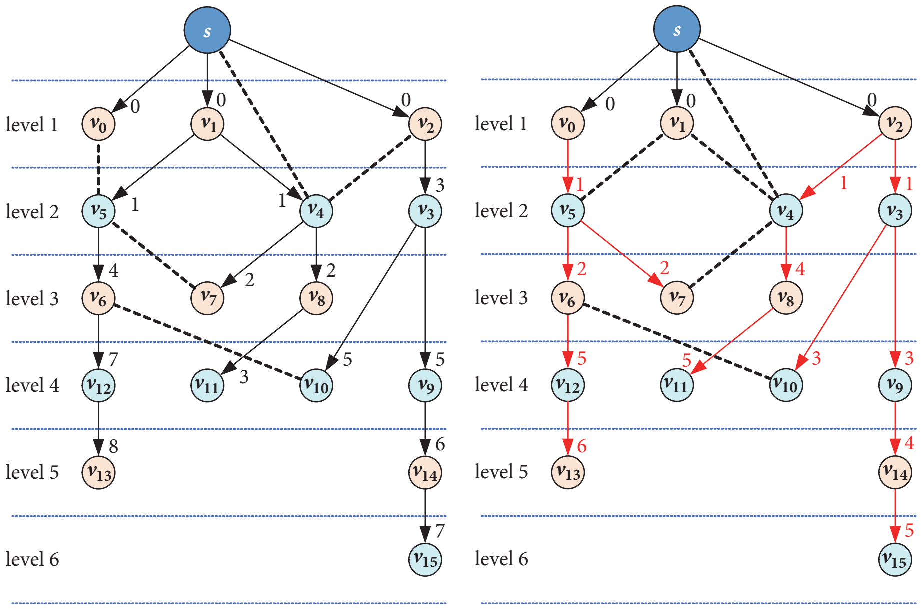
4.1. Overall Idea. Existing broadcast scheduling schemes such as OTAB [21], CLBS [22], and VIA [20] are typically motivated to find a set of forwarders whose simultaneous transmissions will result in as many collision-free receptions



(a) A shortest path tree. A solid link indicates an edge in the tree

(b) Communication graph along with levels of nodes

Figure 3: Level distribution in duty-cycled WSNs.



(a) Degree-based schedule. (latency: 8, number of transmissions: 11)

(b) Critical-path aware schedule. (latency: 6, number of transmissions: 11)

Figure 4: A comparision between a degree-based schedule and a critical-path aware schedule.

as possible. The schemes schedule a broadcast with a preference of transmissions to high-degree nodes. The intuition behind this is that the higher the degree a node has, the more the neighbors it can cover after receiving a message. They use the neighbourhood information of nodes to determine whether a particular node needs to transmit a message. Different transmitting time slots are assigned to nodes having a common neighbor to prevent any collision.

Figure 4(a) illustrates a degree-based broadcast scheduling for the duty-cycled WSN with $T = 2$, shown in Figure 2. In the schedule, the source node s broadcasts a message to its neighbors v_0, v_1, and v_2 at time slot 0. A transmission to v_4 is preferred to be scheduled as the node has the highest degree among the remaining ones. There are four neighboring nodes of v_4 ready to forward the broadcast message. In order to prevent a collision at the common neighbor v_4, only v_1 is

```
Input: $G = (V, E), s \in V, lv(u) (\forall u \in V)$
Output: Broadcast tree $T_B = \{V, E_B\}$
$V_B \longleftarrow \{s\}, E_B \longleftarrow \varnothing$
while $V_B \neq V$ do
    $k \longleftarrow \min\{lv(u) \mid u \in V \setminus V_B\}$
    $L \longleftarrow \{u \in V \mid lv(u) = k\}$                // nodes at level $k$
    while $L \setminus V_B \neq \varnothing$ do
        $p \longleftarrow \arg\max_{u \in \{v \in V_B \mid lv(v) < k\}} |N(u) \cap (L \setminus V_B)|$
        $E_B \longleftarrow E_B \cup \{(p, u) \in E \mid u \in N(p) \cap (L \setminus V_B)\}$
        $V_B \longleftarrow V_B \cup (N(p) \cap L)$
```

Algorithm 1: Degree-Based-SPT.

allowed to forward the message at the time slot 1. The other ones must delay their transmissions to the next working period. As transmission of v_2 is delayed in the example, its neighbor node v_3 receives the message at time slot 3. In the same manner, a transmission from v_5 is delayed to prevent an interference with the transmission of v_4 at time slot 2. As a result, the receiving time of v_6 is postponed to time slot 4. Then, v_6 can forward the message further to its neighbors, and the broadcast completes at time slot 8.

On an SPT rooted at the source node of a duty-cycled WSN, we define a *critical path* as the longest path from the root to the leaves of the tree. For instance, the path ($s - v_2 - v_3 - v_9 - v_{14} - v_{15}$) is a critical path of the network in Figure 2. As node v_3 belongs to the critical path, a delay of transmission from v_2 to v_3 increases the overall broadcast latency. Similarly, if the transmission from v_5 to v_6 is scheduled earlier, it can result in a reduction of the overall broadcast latency. The problem is more severe in highly dense networks because there may be more nodes which cannot be scheduled simultaneously due to collisions. By preferring transmissions to nodes along the critical path, the critical-path aware scheduling selects v_2 to transmit a message to v_3 and v_4 at time slot 1, as shown in Figure 4(b). Consequently, the remaining nodes in the critical path of v_3 can receive the message earlier than in the degree-based schedule. In the same manner, v_5 is selected to forward the message to v_6 and v_7 at time slot 2. Even if a transmission from v_4 to v_8 is delayed to time slot 4, it does not affect the overall broadcast latency. The broadcast completes after time slot 6 with the critical-path aware scheduling.

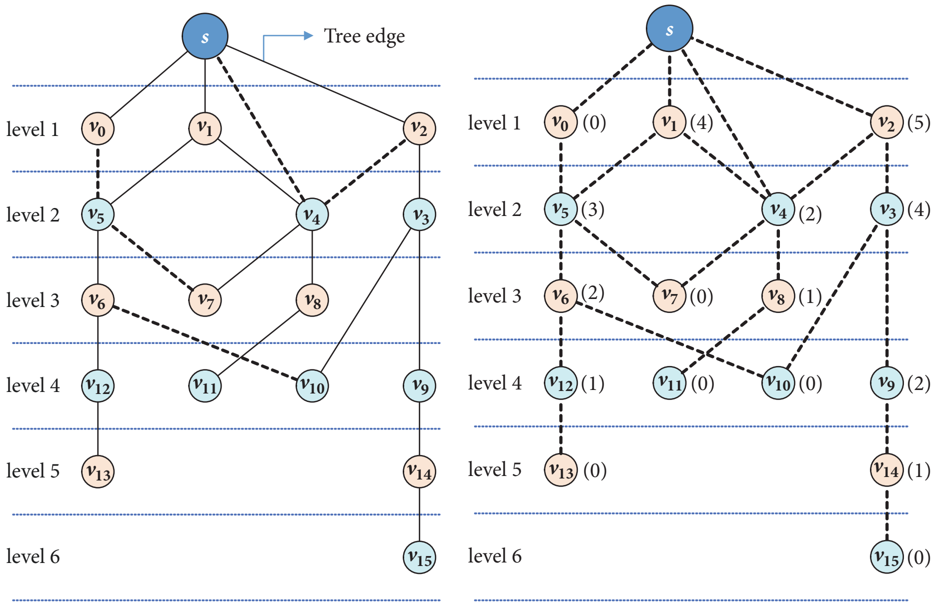
4.2. Criticality Awareness. The critical-path aware scheduling (CAS) approach utilizes node criticality information in its scheduling process to minimize broadcast latency. To this end, all critical paths of a network need to be identified. Note that a critical path is the shortest path from s to a node with maximum level, identified by an SPT rooted at the source node of the network. The proposed SPT construction algorithm connects nodes to the tree in a nondecreasing order of their levels. The algorithm prefers to select high-degree nodes as internal nodes in the tree to reduce the number of transmissions. The notation utilized in this chapter is summarized in Table 2.

Table 2: Notation for the critical-path aware broadcast scheduling.

$G = (V, E)$	communication graph of a network V: set of nodes, E: set of edges
s	source node of the network
T	length of a working period
$N(u)$	1-hop neighbors of node u in G
$R(u, k)$	set of intended receivers with level k of a transmission from node u
$TS(u)$	transmitting schedule of node u
$a(u)$	active slot of node u ($a(u) \in \{0, 1, ..., T - 1\}$)
$lv(u)$	level of node u, i.e. minimum latency to receive a message from s
$la(u)$	latency-ahead value of node u
$parent(u)$	parent node of node u in T_B
$rcv(u)$	receiving time of node u

Given a network $G = (V, E)$ with level of every node and a predefined source node $s \in V$, Algorithm 1 constructs a degree-based SPT rooted at the source node by connecting nodes to the tree level by level. Let V_B denote the set of nodes which have been connected to the tree, initially $V_B = \{s\}$. For adding nodes in each level k to V_B, the algorithm considers nodes in V_B with levels smaller than k as parent candidates. Among the candidates, node p adjacent to the largest number of nodes in $V \setminus V_B$ with the level k is selected as a parent node. Such a node must exist because of the network connectivity. All neighbors of p in $V \setminus V_B$ with the level k become children of p and are to V_B. The process continues until V_B contains all nodes in the network. Obviously, the tree is a shortest path tree rooted at the source node s. Figure 5(a) illustrates a degree-based SPT construction for the network in Figure 2.

To facilitate node criticality awareness, we define the *latency-ahead* value of each node u, denoted by $la(u)$, based on the constructed tree. The latency-ahead value of a node presents the minimum latency for transmitting a message from the node to the farthest leaf node in the subtree rooted at the node. The value is the accumulated costs from the node to a leaf node with the maximum level in its subtree. In other words, a node with a higher latency-ahead value requires a longer time to cover all nodes in its subtree. The


Tree edge
s
level 1
level 2
level 3
level 4
level 5
level 6
v_0 v_1 v_2
v_5 v_4 v_3
v_6 v_7 v_8
v_{12} v_{11} v_{10} v_9
v_{13} v_{14}
v_{15}
v_0 (0) v_1 (4) v_2 (5)
v_5 (3) v_4 (2) v_3 (4)
v_6 (2) v_7 (0) v_8 (1)
v_{12} (1) v_{11} (0) v_{10} (0) v_9 (2)
v_{13} (0) v_{14} (1)
v_{15} (0)

(a) A degree-based shortest path tree

(b) Communication graph along with latency-ahead of nodes

FIGURE 5: Latency-ahead values calculation.

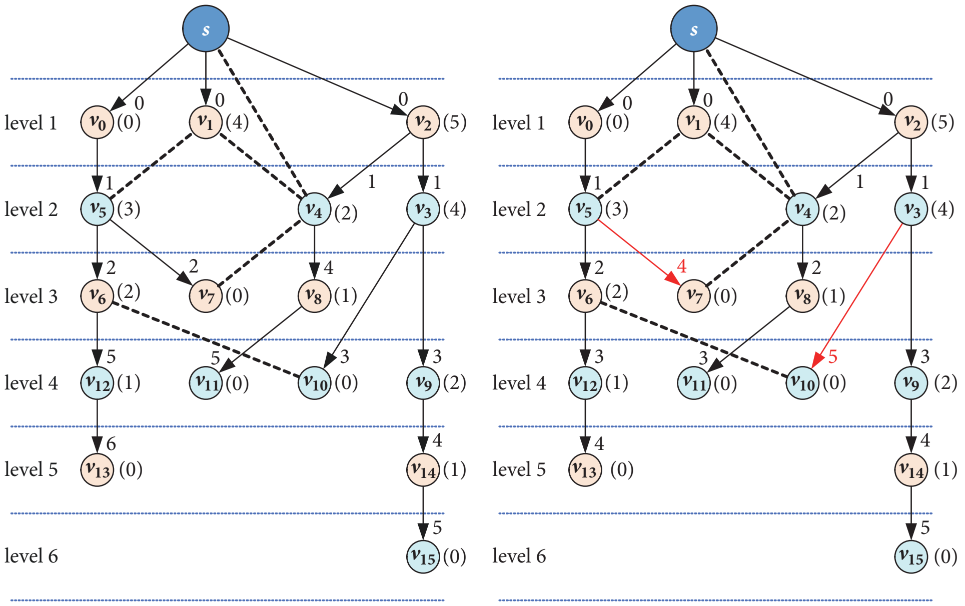
```
Input:  $G = (V, E)$
Output: Transmitting schedule $TS(u)$ for every node $u \in V$
$\forall u \in V, TS(u) \longleftarrow \varnothing, rcv(u) \longleftarrow -\infty$
$rcv(s) \longleftarrow -1$                                   // source node has a message in advance
$i \longleftarrow 0$                                               // current time slot
while $\exists u \in V, rcv(u) = -\infty$ do
    $C \longleftarrow \{u \mid u \in V, rcv(u) \geq -1\}$                 // set of covered nodes
    $\overline{C} \longleftarrow \{u \mid u \in V \setminus C, N(u) \cap C \neq \varnothing, a(u) = i \bmod T\}$   // set of uncovered nodes
    while $C \neq \varnothing$ & $\overline{C} \neq \varnothing$ do
        $u \longleftarrow \underset{u \in \overline{C}}{\operatorname{argmax}}\, la(u)$
        $f \longleftarrow \underset{v \in C \cap N(u)}{\operatorname{argmax}} \left|N(v) \cap \overline{C}\right|$
        $TS(f) \longleftarrow TS(f) \cup \{i\}$
        foreach $v \in N(f) \cap \overline{C}$ do
            $rcv(v) \longleftarrow i, \overline{C} \longleftarrow \overline{C} \setminus \{v\}$
            $C \longleftarrow C \setminus N(v)$
    $i \longleftarrow i + 1$
```

ALGORITHM 2: CF-CAS.

higher latency-ahead value a node has, the more critical the node is. An example of latency-ahead calculation is shown in Figure 5(b).

4.3. Critical-Path Aware Scheduling. In this section, we present two broadcast scheduling schemes employing the critical-path aware scheduling (CAS) to reduce broadcast latency. The proposed schemes, named Collision-Free-CAS (CF-CAS) and Collision-Tolerant-CAS (CT-CAS), follow either collision-free or collision-tolerant strategy, respectively. A node is in a *covered* state if one of its neighbors has been scheduled to transmit a message to the node. Otherwise, it is in an *uncovered* state. The covered node is ready to be scheduled to cover its uncovered neighbors. The schemes schedule a broadcast with a preference of transmissions to uncovered nodes with high latency-ahead values. In each time slot, the schemes minimize the number of transmissions by selecting high-degree covered nodes as forwarders.

4.3.1. CF-CAS. Let C denote the set of covered nodes in a network. Initially, C contains only the source node. The scheduling scheme starts at time slot 0 and iteratively works for each time slot as in Algorithm 2. At the current time slot i ($i \geq 0$), the algorithm searches for the most critical



(a) Collision-free schedule (latency: 6, number of transmissions: 11)

(b) Collision-tolerant schedule (latency: 5, number of transmissions: 13)

Figure 6: An illustration of the critical-path aware scheduling.

uncovered node u that has the highest latency-ahead value and $a(u) = i \bmod T$. All covered neighbors of u are considered as its forwarder candidates. Among the candidates, the algorithm selects node f as a *forwarder* if it covers the largest number of uncovered nodes active at time slot i to reduce the number of transmissions. Let $\overline{C}$ denote the set of uncovered nodes active at the current time slot.

As a transmission from the selected forwarder covers all of its uncovered neighbors active at time slot i; such neighbors, called *listeners*, can receive a message from f at time slot i. The listeners become covered nodes and are excluded from $\overline{C}$. All covered neighbors of the listeners are excluded from C to ensure that they will not be selected as forwarders in time slot i anymore. By doing so, the algorithm prevents any collision at the listeners. CF-CAS continues searching for a node with the highest latency-ahead value in $\overline{C}$ until all nodes active at the current time slot are covered or it cannot find any forwarder for the node. The scheduling algorithm updates the node status, and moves to the next time slot, i.e., $i + 1$, until it covers all nodes in the network, iteratively. Figure 6(a) shows an example of the proposed algorithm on the network shown in Figure 2.

4.3.2. CT-CAS. Observably, a delay of transmission caused by collision-prevention may increase the broadcast latency if the delay occurs along a critical path of the network. With the node criticality awareness, covering nodes in a nonincreasing order of their latency-ahead values can be beneficial for reducing broadcast latency. It motivates us to allow collision at low criticality nodes to speed up the broadcast process for high criticality ones. By doing so, the scheduling accelerates transmissions to nodes with high latency-ahead values. It also increases the number of simultaneous transmissions in each time slot to reduce broadcast latency. The CT-CAS algorithm accommodates the collisions possibly occurring at the low criticality nodes by retransmission to ensure the completion of a broadcast.

Initially, the algorithm offers a preference to the most critical uncovered node u that has the highest latency-ahead value in time slot i. Forwarder f is selected among covered neighbors of u in a similar way as the CF-CAS algorithm. All uncovered neighbors of the selected forwarder are referred to as *listeners* because they can hear a message at time slot i. The listeners are excluded from $\overline{C}$ as they become covered nodes. The difference is that a listener may be changed back to an uncovered state later if it hears more than one message at the current time slot. Let $num(x)$ denote the number of messages that node x can hear in a time slot. The number is increased by one for each listener as in Algorithm 3.

CT-CAS algorithm continues searching for the next critical node u' with the highest latency-ahead value among remaining nodes in $\overline{C}$. To prevent collisions at listeners whose latency-ahead values are not smaller than $la(u')$ and numbers of heard messages are one, all covered neighbors of such listeners are excluded from C. Forwarder f' for node u' is selected in the same manner among the remaining candidates in C. All uncovered neighbors active at time slot i of f' also become listeners. Collisions are allowed at listeners that are common neighbors of f and f'.

The scheduling algorithm iterates the above procedure until all nodes active at the current time slot are covered, or it cannot find any proper forwarder. Only listeners hearing one message in the current tine slot become covered nodes. Other listeners are changed back to an uncovered state as

```
Input: $G = (V, E)$
Output: Transmitting schedule $TS(u)$ for every node $u \in V$
$\forall u \in V, TS(u) \longleftarrow \varnothing, rcv(u) \longleftarrow -\infty, num(u) \longleftarrow 0$
$rcv(s) \longleftarrow -1, num(s) \longleftarrow 1$                // source node has a message in advance
$i \longleftarrow 0$                                              // current time slot
while $\exists u \in V, rcv(u) = -\infty$ do
    $C \longleftarrow \{u \mid u \in V, rcv(u) \geq -1\}$                           // set of covered nodes
    $\overline{C} \longleftarrow \{u \mid u \in V \setminus C, N(u) \cap C \neq \varnothing, a(u) = i \bmod T\}$   // set of uncovered nodes
    $L \longleftarrow \varnothing$                                                  // set of listeners
    while $C \neq \varnothing$ & $\overline{C} \neq \varnothing$ do
        $u \longleftarrow \operatorname{argmax}_{u \in \overline{C}} la(u)$
        $\forall v \in \{x \in L \mid la(x) \geq la(u), num(x) = 1\}, C \longleftarrow C \setminus N(v)$
        $f \longleftarrow \operatorname{argmax}_{v \in C \cap N(u)} \left|N(v) \cap \overline{C}\right|$
        $TS(f) \longleftarrow TS(f) \cup \{i\}$
        foreach $v \in N(f) \cap \overline{C}$ do
            $num(v) \longleftarrow num(v) + 1, \overline{C} \longleftarrow \overline{C} \setminus \{v\}$
            $L \longleftarrow L \cup \{v\}$
    $\forall u \in \{x \in L \mid num(x) = 1\}, rcv(u) \longleftarrow i$
    $\forall u \in \{x \in L \mid num(x) > 1\}, num(u) \longleftarrow 0$
    $i \longleftarrow i + 1$
```

Algorithm 3: CT-CAS.

they cannot receive any message. CT-CAS algorithm resets the number of heard messages for every uncovered node and iteratively moves to the next time slot, $i + 1$, until all nodes in the network are in a covered state. Figure 6(b) shows an example of the proposed algorithm on the network shown in Figure 2.

5. Performance Analysis

Recall that CAS along with collision-free and collision-tolerant strategies is referred to as the CF-CAS scheme and the CT-CAS scheme, respectively. In this section, we estimate approximation ratios between broadcast latencies given by the proposed schemes and the optimal latency for the MLBSDC problem. We first show that approximation ratio of the CF-CAS scheme does not exceed $(\Delta - 1)T$, where Δ is the maximum degree of a communication graph.

The proposed scheme schedules all transmissions based on the latency-ahead value of each node with a preference of node degree. Let d_{CF} denote the broadcast latency in a CF-CAS schedule. Let $R(u, t)$ denote the receiver node set of the scheduled transmission from node u at time slot t, $0 \leq t \leq d_{\mathrm{CF}}$. Note that such receivers are active at the same time slot of $t \bmod T$.

Lemma 1. *In CF-CAS scheme, if a transmission from node u is delayed to prevent a collision at node w then $w \notin R(u, t)$, $\forall t \in [0, d_{\mathrm{CF}}]$.*

Proof. We prove this lemma by contradiction. Assume that a transmission of node u is delayed until time slot t, $0 \leq t \leq d_{\mathrm{CF}}$, to prevent an interference with the prior scheduled transmission of node v at node $w \in R(u, t)$. In the CF-CAS scheme, as v is scheduled prior to u and it has the common uncovered neighbor w, v must have a higher degree than u. According to Algorithm 2, w must be a receiver of v, which contradicts the assumption. □

Let d^* denote the latency of an optimal broadcast schedule for the MLBSDC problem. In the optimal schedule, let R_t^* be the set of all receivers at time slot t, $0 \leq t \leq d^*$. It is worth noting that every node in R_t^* is uncovered and active at time slot t. The following is our key lemma.

Lemma 2. *CF-CAS algorithm takes at most $t(\Delta - 1)T$ time slots to cover all nodes in $\bigcup_{k=0}^{t} R_k^*$, $\forall t \in [0, d^*]$.*

Proof. We prove this lemma by induction on t. As all nodes in level 1 can receive a message from a transmission of s at time slot 0; the claim of this lemma is true for the receiver set R_0^*. Suppose that the claim is true for any receiver set from $\bigcup_{k=0}^{1} R_k^*$ to $\bigcup_{k=0}^{t-1} R_k^*$, where $1 \leq t \leq d^*$. We now prove it for the set $\bigcup_{k=0}^{t} R_k^*$. Let u be a sender at time slot t' that covers some nodes in $R(u, t') \subset \bigcup_{k=0}^{t} R_k^*$. The transmission of node u is delayed till time slot t' to prevent interference with other scheduled transmissions. We denote by $IN(u, t')$ the set of such interfering transmissions. According to the CF-CAS algorithm, we have

$$t' \leq rcv(u) + \left(\left|IN\left(u, t'\right)\right| + 1\right) T \qquad (2)$$

According to Lemma 1, a transmission from node u to nodes in $R(u, t')$ can only be delayed to prevent collision at some uncovered nodes in $N(u) \setminus R(u, t')$. Since the transmission may be delayed until all of such uncovered nodes are covered, we have $|IN(u, t')| \leq |N(u)| - |R(u, t')| - 1$, where 1 accounts for the node $parent(u)$. The reason is that $parent(u)$ has been covered before u and has been excluded from the uncovered node set according to the CF-CAS algorithm. From (2), we have

$$\begin{aligned} t' &\leq rcv(u) + \left(|N(u)| - \left|R\left(u, t'\right)\right| - 1 + 1\right) T \\ &= rcv(u) + \left(|N(u)| - \left|R\left(u, t'\right)\right|\right) T \end{aligned} \tag{3}$$

Noting that in (3) $|N(u)| \leq \Delta$ and $R(u, t') \geq 1$, we have

$$t' \leq rcv(u) + (\Delta - 1) T \tag{4}$$

As u should receive a message before transmitting a message to $\bigcup_{k=0}^{t} R_k^*$, $u \in \bigcup_{k=0}^{i} R_k^*$, where $0 \leq i \leq t-1$. By induction hypothesis, we have

$$rcv(u) \leq i(\Delta - 1) T \leq (t-1)(\Delta - 1) T \tag{5}$$

So the maximum transmitting time slot of node u for nodes in $\bigcup_{k=0}^{t} R_k^*$ can be estimated from (4) and (5) as follows:

$$t' \leq (t-1)(\Delta - 1) T + (\Delta - 1) T = t(\Delta - 1) T \tag{6}$$

Thus, the claim of the lemma holds for the receiver set $\bigcup_{k=0}^{t} R_k^*$, and the proof is complete. □

Theorem 3. *The approximation ratio of the CF-CAS scheme is at most* $(\Delta - 1)T$.

Proof. From the definition of the receiver set in the optimal schedule, we have

$$\bigcup_{k=0}^{d^*} R_k^* = V \setminus \{s\} \tag{7}$$

According to Lemma 2, the maximum time slot to cover all nodes in $\bigcup_{k=0}^{d^*} R_k^*$ is $t \leq (\Delta - 1)Td^*$. Thus, we have $d_{\mathrm{CF}} \leq (\Delta - 1)Td^*$. □

In the following, we estimate the time complexities of the CAS-based schemes.

Theorem 4. *The time complexity of the CF-CAS scheme is* $O(|V|^2 + \Delta^2|V|)$.

Proof. To facilitate the latency-ahead value calculation, Algorithm 1 takes $O(|V|^2)$ time to construct the shortest path tree rooted at the source node. Then, it takes another $O(|V|^2)$ time to assign the latency-ahead value for every node in the network. In total, the time complexity of Algorithm 1 is bounded by $O(|V|^2)$.

The CF-CAS algorithm takes at most $O(|V|)$ time to collect covered set C and uncovered node set $\overline{C}$ in each time slot i, $i \geq 0$. It takes at most $|\overline{C}|$ time to pick the uncovered node with the highest latency-ahead value, and then $|C|$ time to select a parent for the node from set C with a preference of node degree. Recall that uncovered neighbors active at time slot i of the selected parent become listeners at the time slot. The parent iterates at most Δ^2 times to guarantee that all listeners can be covered by preventing their covered neighbors from transmitting a message at time slot i. Thus, the algorithm takes at most $|\overline{C}| + |C| + \Delta^2 = |V| + \Delta^2$ time for each uncovered node. To cover all nodes, the time complexity of the Algorithm 2 is bounded by $O(|V|(|V| + \Delta^2)) = O(|V|^2 + \Delta^2|V|)$. We combine all the running times and conclude that the time complexity of the scheme is $O(|V|^2 + \Delta^2|V|)$. □

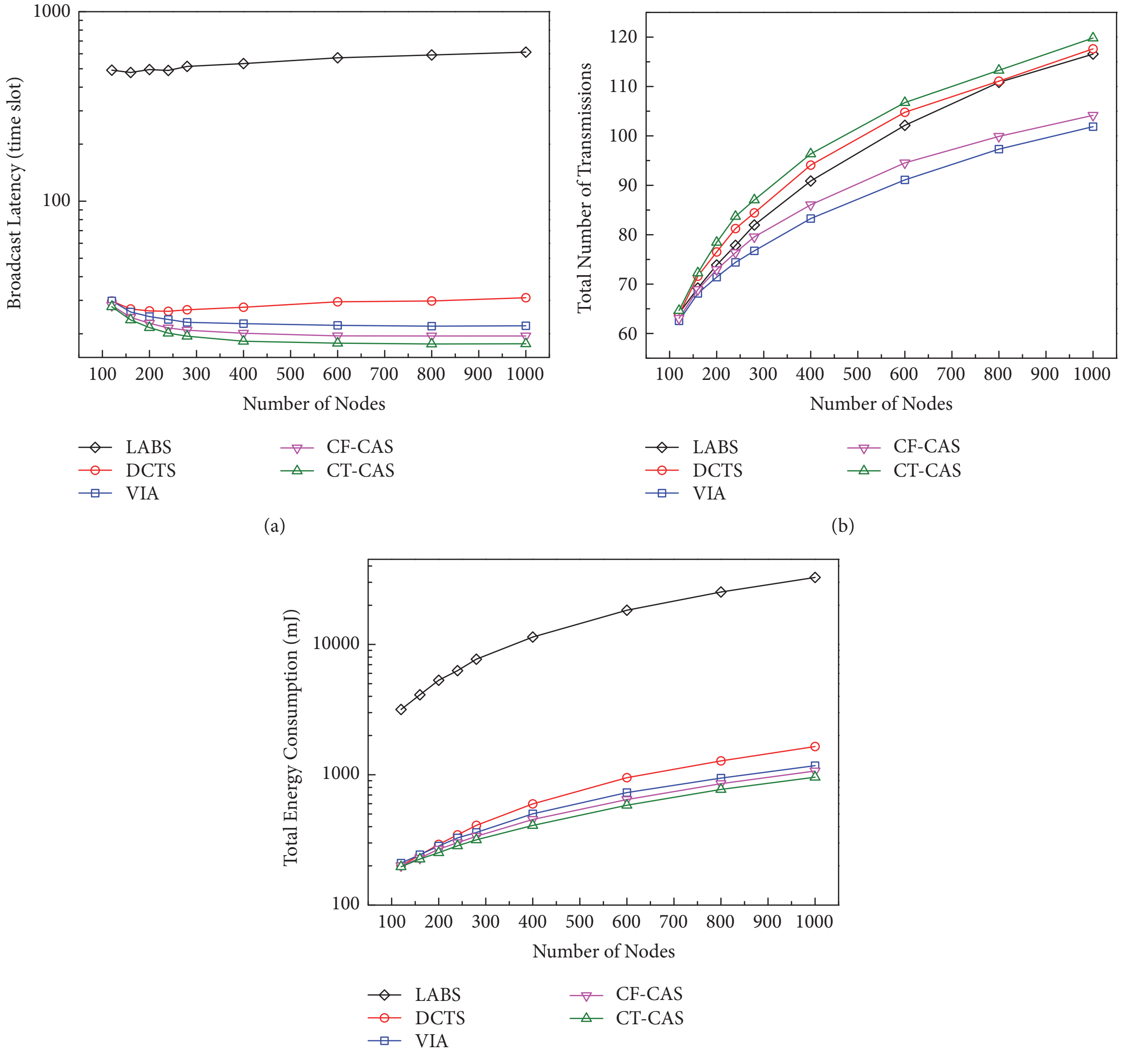
Theorem 5. *The time complexity of the CT-CAS scheme is* $O(\Delta|V|^2)$.

Proof. Similar to the proof of Theorem 4, the CT-CAS scheme takes $O(|V|^2)$ time to assign latency-ahead value for every node in a network. The only difference between two CAS-based schemes is that the CT-CAS algorithm does not require Δ^2 time to guarantee all listeners in a time slot will be covered. Instead, it takes Δ time to collect all listeners and then takes at most $\Delta|V|$ time to prevent collision at nodes with high latency-ahead values. Thus, the algorithm requires $O(|\overline{C}| + |C| + \Delta + \Delta|V|) = O(|V| + \Delta|V|) = O(\Delta|V|)$ time for each uncovered node, as $|V| \geq 1$ and $\Delta \geq 1$. To cover all nodes, the time complexity of Algorithm 3 is bounded by $O(|V|\Delta|V|) = O(\Delta|V|^2)$. Combining this with the time complexity of Algorithm 1, we conclude that the time complexity of the CT-CAS scheme is $O(\Delta|V|^2)$. □

6. Performance Evaluation

6.1. Simulation Environment. In this section, the performance of the proposed schemes is evaluated using a simulator written in C#. The simulation configurations are similar to the ones in [22]. Each network is generated by uniformly deploying all sensor nodes in a square area of 200 m × 200 m. All nodes are assumed to have the same transmission range. They obtain their active time slots randomly. To enable an energy consumption evaluation, we adopt the energy model of the Mica2 platform [43]. A node consumes 9.6 mJ/s and 0.33 mJ/s in an active state and a sleeping state, respectively. It costs 5.76×10^{-3} mJ/byte for transmitting and 0.88×10^{-3} mJ/byte for receiving a message. A message has a fixed size of 28 bytes as in TinyOS. The length of each time slot is 20 milliseconds. We neglect the energy consumed during the network initialization phase.

Broadcast latency, the total number of transmissions, and the total energy consumption of the proposed schemes are compared with those of VIA [20], LABS[22], and DCTS [29]. We did not include ELAC [19] and OTAB [21] into the comparison because their performances in terms of both broadcast latency and number of transmissions are consistently dominated by LABS [22]. The effect of network parameters including network density, transmission range, and the duty-cycle on the performance of these schemes is studied. We vary one of the parameters while fixing the others in each simulation. Each value plotted on the curves is obtained from the results of 200 randomly generated networks with a randomly chosen source node. Table 3 summarizes the simulation parameters.



(a)

(b)

(c)

Figure 7: The impact of the number of nodes (transmission range is 30 m, length of working period is 4).

Table 3: Simulation parameters for the critical-path aware broadcast scheduling.

Parameter	Value
Network area	200 m × 200 m
Energy rate in active	9.6 mJ/s
Energy rate in sleep	0.33 mJ/s
Energy rate in transmitting	5.76×10^{-3} mJ/byte
Energy rate in receiving	0.88×10^{-3} mJ/byte
Message size	28 bytes
Length of time slot	20×10^{-3} s
Number of nodes	120 ~ 1000
Transmission range	20 m ~ 60 m
Length of working period	2 ~ 10 time slots

6.2. Simulation Results

6.2.1. The Impact of the Number of Nodes. Figure 7 presents the performance of the schemes in low-density networks and high-density ones where the number of nodes is varied from 120 to 280 and from 400 to 1000 with increments of 40 and 200, respectively. The transmission range and the length of the working period are fixed to 30 m and 4, respectively. In low-density scenarios, broadcast latencies of all schemes decrease when the network size increases because a node has more choices to forward a message, as shown in Figure 7(a). When the network density becomes high, there are more levels in a network. As level-based schemes, LABS and DCTS require more time to complete a broadcast process. The broadcast latencies of VIA and two proposed schemes decrease slightly since a node needs to wait longer to prevent interference with its neighbors. In all cases, the proposed schemes consistently outperform in terms of broadcast latency. By accelerating transmissions for nodes with high criticality, the broadcast latency of CF-CAS is $5.3 - 12$ percent shorter than that of VIA. CT-CAS further reduces the latency up to 9.3 percent more than CF-CAS by allowing collision at less critical nodes.

Figure 7(b) shows that the total number of transmissions of all schemes grows. The reason is that each forwarding node has more neighbors when the network density increases. It leads to the increment of number of transmissions to cover the neighbors. Consequently, the total energy consumption of all the schemes increases as shown in Figure 7(c). In order to reduce broadcast latency, CF-CAS and CT-CAS require up to 3.8 percent and 17.6 percent more transmissions than VIA, respectively. DCTS and CT-CAS produce more transmissions than LABS and CF-CAS because of additional transmissions for nodes experiencing collisions. Thanks to the shorter broadcast schedules, all nodes in a network can stop working earlier in the proposed schemes. It results in an improvement of up to 19.9 percent in total energy consumption compared to VIA.

The experiment results depict a notable extra delay of LABS because it unnecessarily suspends the collision-free transmissions to nonindependent nodes in a layer until all independent nodes in the layer received a message. This extra latency persistently results in bigger broadcast latencies than those of other schemes throughout all our experiments. To reflect the influence of CAS in reducing broadcast latency, the rest of comparisons only include VIA and DCTS, which produce the shortest broadcast latency among existing schemes deploying a collision-free and a collision-tolerant approaches, respectively.

6.2.2. The Impact of the Transmission Range. The effect of varying transmission ranges on the performance of the schemes is studied with 400 nodes. The length of the working period is fixed to 4. As coverage area of a node gets bigger when its transmission range increases, a broadcast message can reach to a further node in a transmission. In other words, the message propagation may be faster with the bigger transmission range. This leads to a reduction of the broadcast latencies of all the schemes, as shown in Figure 8(a). CF-CAS achieves from 2.9 to 12 percent latency reduction over VIA while CT-CAS reduces up to 9.3 percent more.

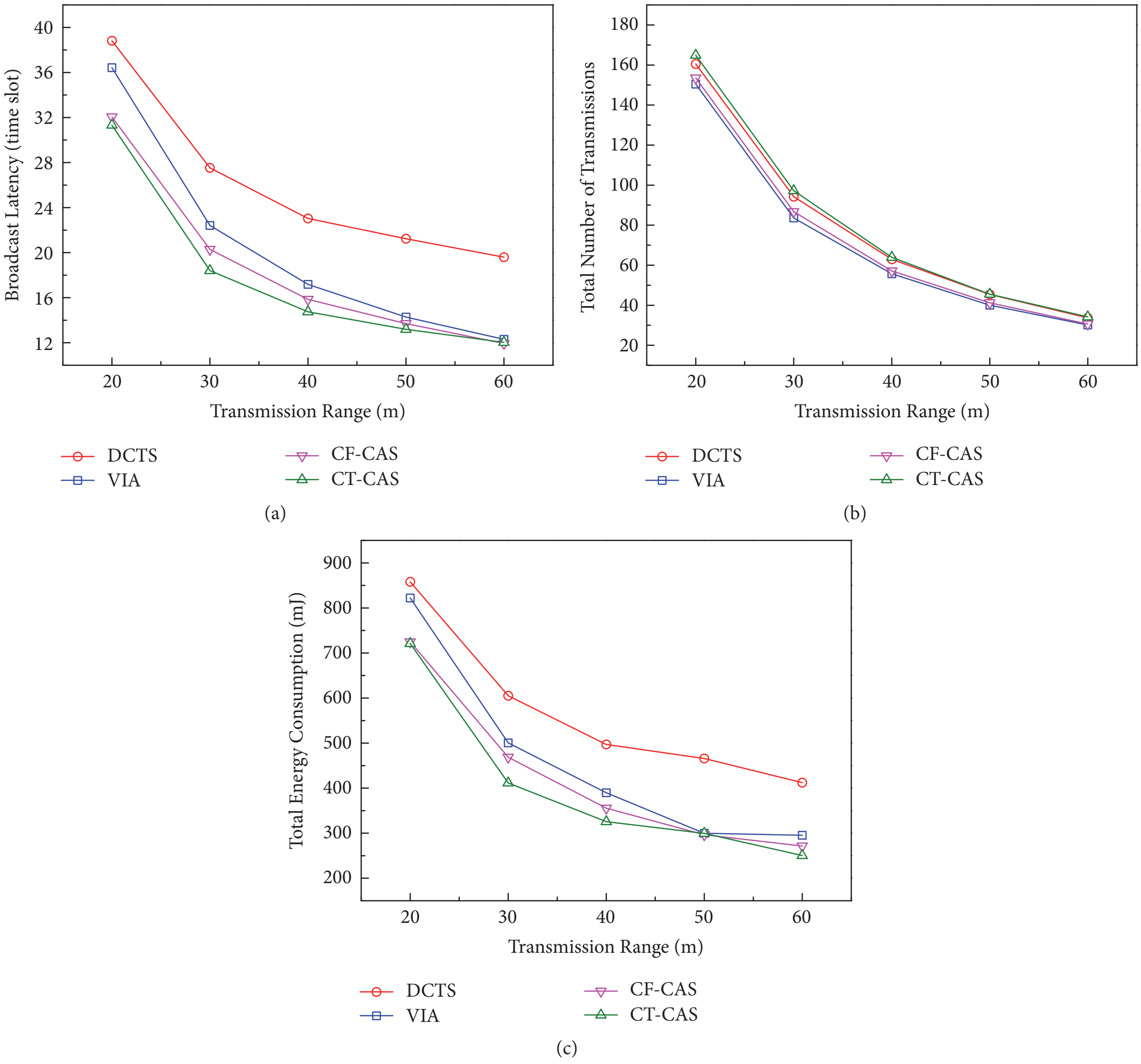
Subsequently, the number of neighbors of a node increases as its transmission range does. As the node can cover more neighbors with a transmission, it potentially requires a fewer number of transmissions to cover all nodes in a network. The total number of transmissions and the total energy consumption decrease, as shown in Figures 8(b) and 8(c), respectively. Due to additional transmissions to nodes experiencing collision, CT-CAS requires $7.4 - 11.9$ percent more transmissions than CF-CAS. The two schemes produce at most 16.3 and 3.8 percent more transmissions than VIA, respectively. However, the total energy consumptions of CF-CAS and CT-CAS consistently outperform VIA by $1.1 - 11.8$ and $0.1 - 17.7$ percent due to their shorter broadcast schedules, respectively.

6.2.3. The Impact of the Duty-Cycle. Figure 9 presents the impact of the duty-cycle on the performance of the schemes. While varying the length of the working period, we fix the number of nodes at 400 and the transmission range at 30 m. As nodes sleep longer when the length of working period increases, a forwarding node takes more time to wait for its receivers waking up. Figure 9(a) shows that the broadcast latencies of all schemes increase. Thanks to node criticality awareness, CF-CAS improves the broadcast latency by $1.8 - 15$ percent compared to VIA. Instead of deferring a transmission to prevent a collision at every node, CT-CAS allows collisions at some nodes to reduce the broadcast latency by $0.5 - 10.2$ percent compared to CF-CAS.

In a longer working period, a forwarding node may require more transmissions to cover all of its neighbors with different active slots. Thus, the total number of transmissions grows with the decrease of working period length, as shown in Figure 9(b). However, the total energy consumption of VIA, CF-CAS, and CT-CAS decreases because their broadcast schedules are short enough and there will be more nodes sleeping in each working period with a bigger T, as shown in Figure 9(c). Although the proposed schemes produce $2.4 - 28.8$ percent more transmissions than VIA, CF-CAS and CT-CAS, respectively, require $0.8 - 14.8$ and $9.9 - 20.8$ percent less energy by completing a broadcast faster.

6.2.4. The Trade-Off between Number of Transmissions and Broadcast Latency. While the network diameter, i.e., the shortest distance between the two most distant nodes, increases by adding more nodes or reducing the transmission range, the broadcast latency improvements of CF-CAS and CT-CAS schemes compared to VIA also increase, as shown in Figures 7(a) and 8(a). This reveals that CAS approach reduces more broadcast latency in networks with longer diameters. By contrast, the proposed schemes become worse in terms of number of transmissions in such networks, as shown in Figures 7(b) and 8(b). The reason is that transmissions from high-degree forwarding nodes in VIA scheme can cover more nodes than transmissions from nodes in critical paths in the proposed schemes.

To present the trade-off between the number of transmissions and the latency of a broadcast, we refer to nodes with latency-ahead values higher than $\tau(0 \leq \tau \leq 1)$ times the maximum latency-ahead value as *critical nodes* in the network. The other nodes are referred to as noncritical ones. We slightly modify the CT-CAS algorithm, later on referred to as CT-CAS(τ), by allowing collisions at only nocritical nodes. After covering a critical node in a time slot, its covered neighbors will not be selected as forwarders in the time slot, i.e., excluded from the covered node set, to prevent collision at the critical node. It is worth noting that the CT-CAS algorithm is the case of CT-CAS(τ) when $\tau = 1$. The bigger the parameter τ is, the less the critical nodes a network has. And thus, it increases the number of collisions possible occurring at noncritical nodes.



(a)

(b)

(c)

FIGURE 8: The impact of the transmission range (number of nodes is 400, length of working period is 4).

From simulation results shown in Figure 10, we observe that the increase of τ results in performance improvement in terms of broadcast latency. With a high τ, transmissions to nodes with high latency-ahead values receive more preference in a broadcast schedule. Such nodes can receive a message sooner to forward the message further. Hence, the overall broadcast latency is reduced. Obviously, more transmissions are required to accommodate collision possibly occurring at the noncritical nodes. With $\tau = 1$, the latency of CT-CAS(1) is the closest to the lower bound of the broadcast latency, which is the the maximum latency-ahead value in a network. The results show that the broadcast latency of CT-CAS(1) is at least 4.1 percent lower than that of CF-CAS. CT-CAS(1) requires at most 14.3 percent more transmissions than CF-CAS due to retransmissions.

7. Conclusions

This paper presents a critical-path aware scheduling (CAS) for the Minimum Latency Broadcast Scheduling problem in duty-cycled wireless sensor networks. The scheduling reduces the broadcast latency by preferring transmissions to nodes along the critical paths of a network. We employ the scheduling in two latency-efficient broadcast schemes: Collision-Free-CAS (CF-CAS) and Collision-Tolerant-CAS (CT-CAS). By allowing collisions at low criticality nodes, CT-CAS accelerates the broadcast process for high criticality ones. It produces a shorter broadcast schedule with more transmissions than CF-CAS. Both proposed schemes improve up to 19.6 percent in terms of broadcast latency while increasing the number of transmissions at most 28.8 percent compared with those of existing schemes. The shorter broadcast schedule reduces up to 20.8 percent the total energy consumption.

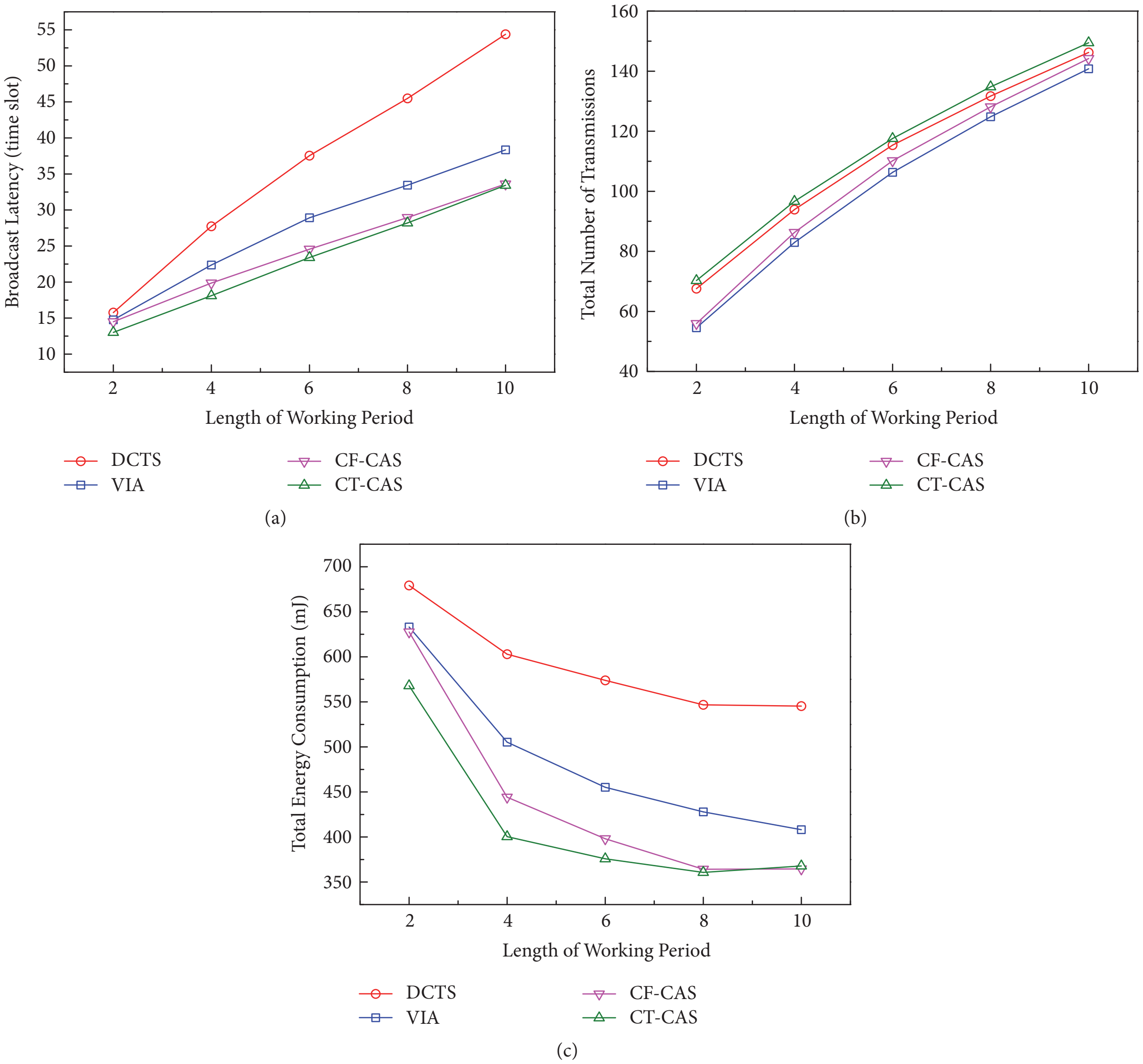


Figure 9: The impact of the duty-cycle (number of nodes is 400, transmission range is 30).

For further study, finding a solution for latency-efficient multicasting in duty cycled WSNs is an appealing problem. Broadcast schemes that balance energy consumption of every node in order to maximize the network lifetime is another challenging problem. Last but not least, we intend to study more realistic uneven node deployments to prove the applicability of our proposed methods.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

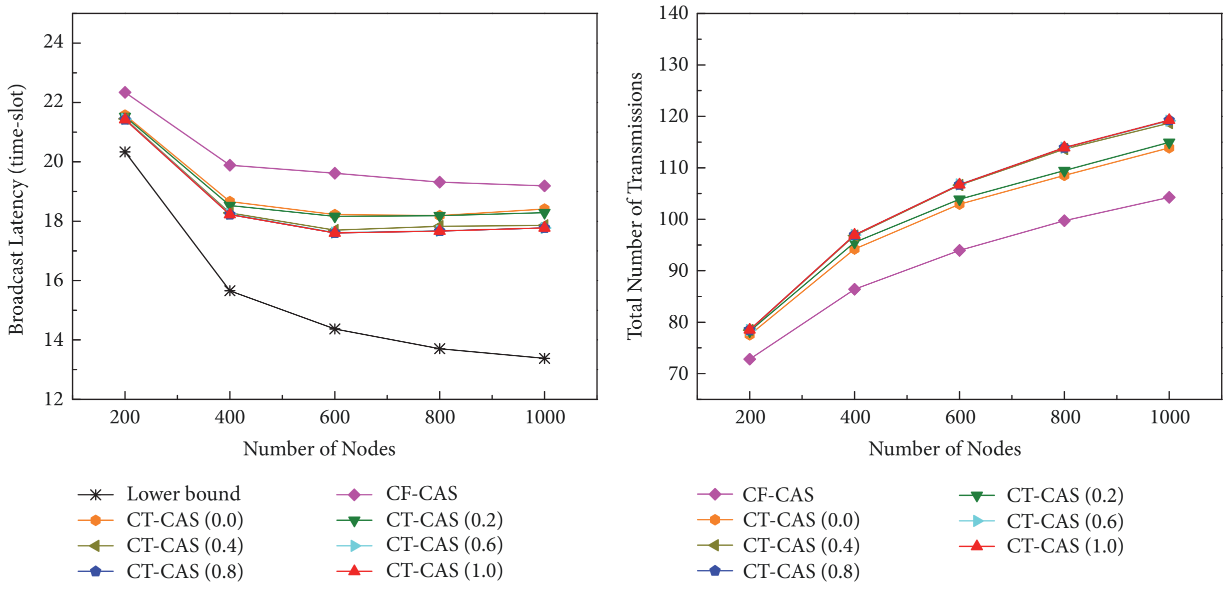
This paper is an extended version of our paper published in International Conference on Parallel and Distributed Processing Techniques and Applications 2015, Duc-Tai Le, Thang Le Duc, Vyacheslav V. Zalyubovskiy, Dongsoo S. Kim, and Hyunseung Choo: On Minimizing Broadcast Latency in Duty-cycled Wireless Sensor Networks (Proceeding of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2015), 2015, pp. 485-490).

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

This paper has been primarily conducted by Duc-Tai Le under the supervision of Hyunseung Choo. Giyeol Im, Thang Le Duc, Vyacheslav V. Zalyubovskiy, and Dongsoo S.



(a)

(b)

FIGURE 10: The trade-off between the number of transmissions and the latency of a broadcast (transmission range is 30 m, the length of working period is 4).

Kim contributed to the descriptions and discussions on the proposed schemes and the simulation results presented in the paper.

Acknowledgments

This research was supported in part by Korean government, under IITP (B0101-15-1366), G-ITRC (IITP-2016-R6812-16-0001), PRCP (NRF-2010-0020210), and BSRP (NRF-2016R1D1A1B03934660), respectively.

References

[1] P. Coy and N. Gross, "21 Ideas for the 21st Century," in *Business Week Online*, pp. 78–167, 1999.

[2] C. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[3] K. Suto, H. Nishiyama, N. Kato, and CW. Huang, "An energy-efficient and delay-aware wireless computing system for industrial wireless sensor networks," *IEEE Access*, vol. 3, pp. 1026–1035, 2015.

[4] C. H. W. Oey and S. Moh, "A survey on temperature-aware routing protocols in wireless body sensor networks," *Sensors*, vol. 13, no. 8, pp. 9860–9877, 2013.

[5] Y. Gao, W. Cheng, H. Zhang, and Z. Li, "Heterogeneous Statistical QoS Provisioning over Wireless Powered Sensor Networks," *IEEE Access*, vol. 5, pp. 7910–7921, 2017.

[6] B. Sabin and M. Sangman, "A survey of MAC protocols for cognitive radio body area networks," *Sensors*, vol. 15, no. 4, pp. 9189–9209, 2015.

[7] C. Xia, X. Jin, L. Kong, and P. Zeng, "Bounding the Demand of Mixed-Criticality Industrial Wireless Sensor Networks," *IEEE Access*, vol. 5, pp. 7505–7516, 2017.

[8] J. Zhu, Y. Zou, and B. Zheng, "Physical-Layer Security and Reliability Challenges for Industrial Wireless Sensor Networks," *IEEE Access*, vol. 5, pp. 5313–5320, 2017.

[9] N. Li, J. Martínez, J. Meneses Chaus, and M. Eckert, "A Survey on Underwater Acoustic Sensor Network Routing Protocols," *Sensors*, vol. 16, no. 3, p. 414, 2016.

[10] B. Tian, K. M. Hou, and H. Zhou, "The traffic adaptive data dissemination (TrAD) protocol for both urban and highway scenarios," *Sensors*, vol. 16, no. 6, 2016.

[11] Xu Luo and Jun Yang, "Water Pollution Detection Based on Hypothesis Testing in Sensor Networks," *Journal of Sensors*, vol. 2017, Article ID 3829894, 8 pages, 2017.

[12] H. Salarian, K.-W. Chin, and F. Naghdy, "Coordination in wireless sensor-actuator networks: a survey," *Journal of Parallel and Distributed Computing*, vol. 72, no. 7, pp. 856–867, 2012.

[13] R. Gandhi, A. Mishra, and S. Parthasarathy, "Minimizing broadcast latency and redundancy in ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 840–851, 2008.

[14] S. C.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, "Minimum-latency broadcast scheduling in wireless ad hoc networks," in *Proceedings of the IEEE 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 733–739, May 2007.

[15] R. Gandhi, Y.-A. Kim, S. Lee, J. Ryu, and P.-J. Wan, "Approximation algorithms for data broadcast in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 7, pp. 1237–1248, 2012.

[16] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[17] S. He, J. Chen, D. K. Y. Yau, H. Shao, and Y. Sun, "Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1090–1102, 2012.

[18] Q. Yang, S. He, J. Li, J. Chen, and Y. Sun, “Energy-efficient probabilistic area coverage in wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, 2014.

[19] J. Hong, J. Cao, W. Li, S. Lu, and D. Chen, “Sleeping schedule-aware minimum latency broadcast in wireless ad hoc networks,” in *Proceedings of the IEEE International Conference on Communications (ICC ’09)*, June 2009.

[20] Y. Duan, S. Ji, and Z. Cai, “Generalized broadcast scheduling in duty-cycle multi-hop wireless networks,” in *Proceedings of the 2011 IEEE 30th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–6, Orlando, FL, USA, November 2011.

[21] X. Jiao, W. Lou, J. Ma, J. Cao, X. Wang, and X. Zhou, “Minimum latency broadcast scheduling in duty-cycled multihop wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 110–117, 2012.

[22] D. T. Le, T. Le Duc, V. V. Zalyubovskiy, D. S. Kim, and H. Choo, “LABS: Latency aware broadcast scheduling in uncoordinated Duty-Cycled Wireless Sensor Networks,” *Journal of Parallel and Distributed Computing*, vol. 74, no. 11, pp. 3141–3152, 2014.

[23] S. Guo, L. He, Y. Gu, B. Jiang, and T. He, “Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links,” *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2787–2802, 2014.

[24] X. Jiao, W. Xiao, B. Ge, and Y. Chen, “On Minimum-Latency Broadcast in Multichannel Duty-Cycled Wireless Sensor Networks,” *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

[25] D. T. Nguyen, J. Choe, T. L. Duc, D. T. Le, V. V. Zalyubovskiy, and H. Choo, “Delay-sensitive flooding based on expected path quality in low duty-cycled wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, 2016.

[26] L. Xu, X. Zhu, H. Dai, X. Wu, and G. Chen, “Towards energy-fairness for broadcast scheduling with minimum delay in low-duty-cycle sensor networks,” *Computer Communications*, vol. 75, pp. 81–96, 2016.

[27] D. T. Le, *Latency Aware Broadcast Scheduling for the Internet of Things [Ph.D. dissertation]*, Sungkyunkwan University, 2016.

[28] R. Zhang, H. Moungla, J. Yu, L. Chen, and A. Mehaoua, “Multichannel Broadcast in Duty-Cycling WBANs via Channel Hopping,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2351–2361, 2017.

[29] D. T. Le, T. Le Duc, V. V. Zalyubovskiy, D. S. Kim, and H. Choo, “Collision-tolerant broadcast scheduling in duty-cycled wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 100, pp. 42–56, 2017.

[30] L. Cheng, J. Niu, C. Luo et al., “Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks,” *Computer Networks*, vol. 134, pp. 66–77, 2018.

[31] J. Hong, J. Cao, W. Li, S. Lu, and D. Chen, “Minimum-transmission broadcast in uncoordinated duty-cycled wireless Ad Hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 307–318, 2010.

[32] B. Tang, B. Ye, J. Hong, K. You, and S. Lu, “Distributed low redundancy broadcast for uncoordinated duty-cycled WANETs,” in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference: “Energizing Global Communications”, GLOBECOM 2011*, December 2011.

[33] T. Le Duc, D. T. Le, V. V. Zalyubovskiy, D. S. Kim, and H. Choo, “Level-based approach for minimum-transmission broadcast in duty-cycled wireless sensor networks,” *Pervasive and Mobile Computing*, vol. 27, pp. 116–132, 2016.

[34] T. L. Duc, D. T. Le, V. V. Zalyubovskiy, D. S. Kim, and H. Choo, “Towards broadcast redundancy minimization in duty-cycled wireless sensor networks,” *International Journal of Communication Systems*, vol. 30, no. 6, p. e3108, 2017.

[35] M. Miller, C. Sengul, and I. Gupta, “Exploring the Energy-Latency Trade-Off for Broadcasts in Energy-Saving Sensor Networks,” in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS’05)*, pp. 17–26, Columbus, OH, USA.

[36] C. Sengul, I. Gupta, and M. J. Miller, “Adaptive probability-based broadcast forwarding in energy-saving sensor networks,” *ACM Transactions on Sensor Networks*, vol. 4, no. 2, pp. 1–32, 2008.

[37] G. Im, D. T. Le, H. Choo, and D. S. Kim, “Critical-path aware broadcast scheduling in duty-cycled Wireless Sensor Networks,” in *Proceedings of the 2015 International Conference on Information Networking (ICOIN)*, pp. 410-411, January 2015.

[38] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[39] D. T. Le, T. Le-Duc, V. Zalyubovskiy, D. S. Kim, and H. Choo, “On minimizing broadcast latency in duty-cycled wireless sensor networks,” in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2015)*, pp. 485–490, 2015.

[40] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *SIGOPS Operating Systems Review*, vol. 36, pp. 147–163, 2002.

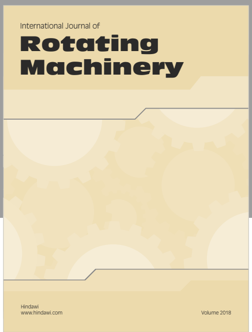
[41] S. Yoon, C. Veerarittiphan, and M. L. Sichitiu, “Tiny-sync: Tight time synchronization for wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 3, no. 2, 2007.

[42] X. Cao, F. Yang, X. Gan et al., “Joint estimation of clock skew and offset in pairwise broadcast synchronization mechanism,” *IEEE Transactions on Communications*, vol. 61, no. 6, pp. 2508–2521, 2013.

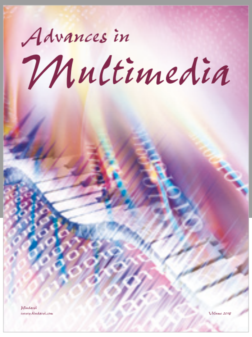
[43] B. Jiang, B. Ravindran, and H. Cho, “Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 735–747, 2013.

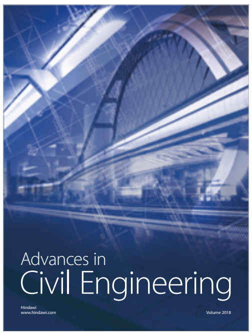
Journal of
Engineering

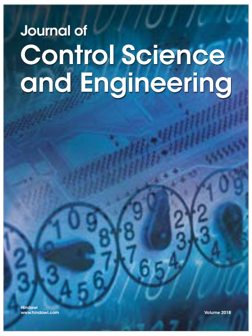
The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

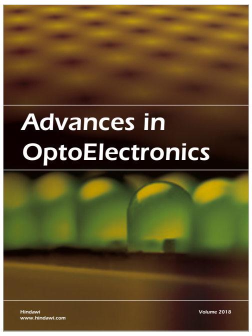

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

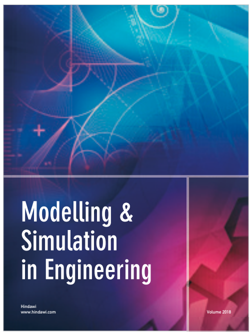
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

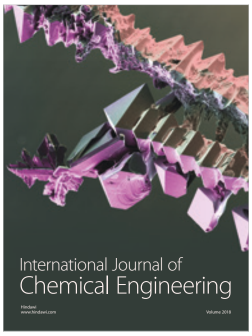

Advances in
OptoElectronics

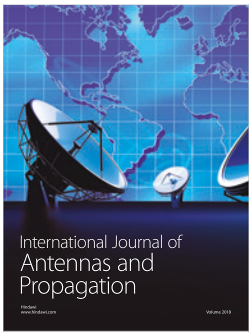
VLSI Design

International Journal of
Navigation and
Observation

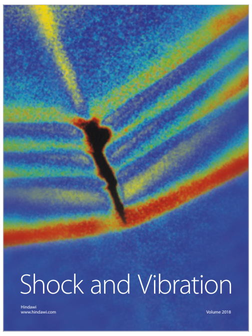

Modelling &
Simulation
in Engineering

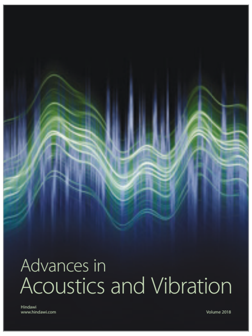
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration