Computer Networks 136 (2018) 155–170

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Achieving adaptive broadcasting performance tradeoff for energy-critical sensor networks: A bottom-up approach

Lijie Xu [a,b,c,*], Geng Yang [a,b], Jia Xu [a,b], Lei Wang [a,b], Haipeng Dai [c]

[a] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, PR China
[b] Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing, Jiangsu 210023, PR China
[c] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210023, PR China

ARTICLE INFO

Article history:
Received 15 September 2017
Revised 26 January 2018
Accepted 12 March 2018
Available online 13 March 2018

Keywords:
Wireless sensor networks
Low-duty-cycle
Broadcasting schedule
Bottom-up solution
Adaptive broadcasting performance tradeoff

ABSTRACT

Low-duty-cycle mode is widely adopted in energy-critical wireless sensor networks (WSNs). Such mode greatly reduces the energy waste caused by idle listening. However, it brings many new challenges for broadcasting. This paper mainly focuses on the minimum cost broadcast problem for low-duty-cycle WSNs. We propose a novel opportunistic broadcasting transmission model, which makes full use of the broadcast nature of wireless media to reduce the total energy consumption for broadcasting. The key idea is to allow nodes to defer their wake-up slots to opportunistically overhear the broadcasting messages sent by their neighbors, which could reduce the total energy consumption for broadcasting but increase the average end-to-end broadcasting delay. In this paper, we define a generalized broadcasting cost function, which can make a flexible tradeoff between average end-to-end broadcasting delay and total energy consumption for broadcasting, to adaptively meet various broadcasting performance requirements. Our target is to utilize the opportunistic broadcasting transmission model to design an efficient broadcasting schedule for low-duty-cycle WSNs, so that the broadcasting cost function is minimized. First, we define the Receiver-Constrained Minimum Cost Single-hop Broadcast Problem (RC-MCSB) and propose an optimal solution with a polynomial running time. Next, we extend the solution of RC-MCSB problem to our target problem and present a novel and efficient bottom-up solution. The simulation results have verified the significant performance advantage of our proposed bottom-up solution over the existing top-down solutions and the other solutions.

1. Introduction

As the key technique of Internet of Things [1,2], wireless sensor networks (WSNs) [3] have received lots of attentions in the past decades and made great progress in both academic and industrial communities. Many typical WSNs applications [4–6] often require nodes should be deployed in tough environments where the sensor nodes are difficult to replace or recharge their batteries, and also require the network system should run for a long enough period. Therefore, how to efficiently prolong the network lifetime becomes a very important problem. To this end, many energy efficient solutions have been proposed for energy-critical WSNs [7–10]. Most of these existing works assume the network is operated at low-duty-cycle mode, where each sensor node has its own working-sleeping schedule to wake up periodically. Low-duty-cycle operation can significantly reduce the energy consumption caused by idle listening, which has been verified as the main source of energy waste for WSNs with low traffic characteristic [11]. However, such mode still brings many new challenges, especially for broadcasting applications [12], which are the often-used fundamental functions for WSNs.

First, low-duty-cycle mode will have an effect on delay performance for broadcasting. Specifically, each sender cannot forward the broadcasting message until the receiver wakes up, it will result in a notable increase on communication delay between any neighboring nodes, which is called sleep latency [13]. More importantly, the energy performance for broadcasting could also be significantly degraded. For the traditional always-awake WSNs, the implementation of any local single-hop broadcast just requires one transmission from the sender, due to the inherent broadcast nature of wireless media. However, low-duty-cycle mode will make the neighboring nodes have totally different working-sleeping schedules such that wireless media could lose its inherent advantage

* Corresponding author at: School of Computer Science, Nanjing University of Posts and Telecommunications, 9 Wenyuan Road, Qixia District, Nanjing, 210023, PR China

E-mail addresses: ljxu@njupt.edu.cn (L. Xu), yangg@njupt.edu.cn (G. Yang), xujia@njupt.edu.cn (J. Xu), leiwang@njupt.edu.cn (L. Wang), dhpphd2003@gmail.com (H. Dai).

https://doi.org/10.1016/j.comnet.2018.03.007
1389-1286/

for broadcasting, which implies any local single-hop broadcast will be inefficiently implemented by multiple unicasts. For broadcasting applications with large messages, such as code update, the degradation on energy performance for broadcasting will be more significant. Thus, how to design a broadcast scheme with high performance for low-duty-cycle WSNs is an important and challenging issue.

For real broadcasting applications, delay and energy are usually both the important performance metrics. Many existing works have investigated the energy optimization problem for broadcasting under delay constraints. However, it is usually unnecessary to require the broadcasting should have a strict delay constraint for many real applications. More broadcasting applications focus on the tradeoff between delay performance and energy performance. In practice, broadcasting performance requirements are usually application-specific. For example, the broadcasting applications with small and urgent messages, such as *configuration dissemination*, will pay more attention on delay performance than energy performance, in order to satisfy the new updated system requirement as soon as possible and reduce the chance of false positive or false negative. The broadcasting applications with large and non-urgent messages, such as *code update*, will pay more attention on energy performance than delay performance, since the updated code image normally consists of multiple packets, which will further deteriorate the energy efficiency of broadcasting. For broadcast problem in low-duty-cycle WSNs, thus, we tend to define the optimization objective as the generalized broadcasting performance, which can make an adaptive tradeoff between delay performance and energy performance. On the other hand, most of the existing works assume the traditional transmission model for broadcasting, in which every node will receive the broadcasting message at its scheduled wake-up time. Such model will lead to a low energy efficiency for broadcasting, since it totally ignores the inherent broadcast nature of wireless media and any local single-hop broadcast will be realized by a number of unicasts. Actually, we find that even for low-duty-cycle WSNs, the inherent broadcast nature of wireless media can still be fully exploited to improve the energy efficiency for broadcasting, by adopting a novel opportunistic broadcasting transmission model. The key idea is to allow any node to defer its wake-up time to opportunistically overhear the broadcasting message from the nearby forwarder. By such way, we can find the total energy consumption for broadcasting could be reduced, however, at the cost of the increase of average broadcasting delay. In other words, the opportunistic broadcasting transmission model can essentially provide a flexible control on the tradeoff between delay performance and energy performance.

Similar to the literature [39], we employ broadcasting cost, which is defined as the function weights delay and energy, to characterize the generalized broadcasting performance in this paper. Note that, we can adaptively adjust the tradeoff factor parameter in the broadcasting cost function to characterize various broadcasting performance requirements. This paper aims to utilize the opportunistic broadcasting transmission model to design an efficient broadcasting schedule for low-duty-cycle WSNs, so that the broadcasting cost function is minimized.

The main contributions of our work are summarized as follows:

- Compared with most of the existing works, the optimization objective of this paper is more practical. We define a generalized broadcasting cost function, which can provide a flexible control on the tradeoff between average end-to-end broadcasting delay and total energy consumption for broadcasting. By adaptively adjusting the tradeoff factor in objective function, our proposed solution can be universally applicable for the applications with various broadcasting performance requirements.
- We adopt the novel opportunistic broadcasting transmission model. By allowing nodes to postpone their wale-up time to opportunistically overhear the message, such model provides a flexible control on the tradeoff between delay performance and energy performance, which can offer a much more fine-grained design for the optimal broadcasting schedule to the target problem.
- Different from our previous work [14], in this paper, we come up with a novel bottom-up approach to address our target problem. Specifically, we first define a Receiver-Constrained Minimum Cost Single-hop Broadcast Problem (RC-MCSB), and propose an efficient dynamic programming algorithm with a polynomial running time. Then, we extend the solution of RC-MCSB to our target problem, *i.e.*, the Minimum Cost Broadcast Problem (MCB) for multihop networks, and devise a novel and efficient bottom-up algorithm. Further, we discuss how to extend our proposed algorithm to the generalized case where a few neighboring nodes could have the identical working-sleeping schedule.
- Extensive simulation results show that our proposed bottom-up algorithm has a much better performance than the existing top-down solutions and the other solutions.

The rest of the paper is organized as follows: Section 2 summarizes the related work. Section 3 illustrates the network model and states the problem. Detailed description and analysis of our proposed algorithm are presented in Section 4. Followed by the discussion and the simulation results in Section 5 and Section 6. Finally, Section 7 concludes our findings.

2. Related work

As an important and challenging issue, the broadcast problem for low-duty-cycle WSNs has been well-investigated by the researchers in the past years [12,15–38]. The existing works always regard delay and energy efficiency as the main broadcasting performance metrics.

Guo et al. [15] designed an opportunistic flooding algorithm for low-duty-cycle WSNs with unreliable links. By letting senders make probabilistic forwarding decisions based on the delay distribution of next-hop nodes, the opportunistic flooding algorithm greatly improves the broadcasting delay. Lu and Whitehouse [16] utilized the capture effect in physical layer to propose an efficient broadcasting protocol for low-duty-cycle WSNs, the solution allows the concurrent transmissions between multiple nodes and thus greatly reduces the broadcasting delay. In [19], the authors proposed a completely contention-free data dissemination protocol, *i.e.*, Pando, which can continuously disseminate rateless encoded packets over the parallel pipelines by integrating Fountain codes with constructive interference and pipelining. The experimental results reveal that Pando can provide 100% reliability and significantly reduce the dissemination delay. Zhang et al. [20] considered the broadcast problem for multi-channel asymmetric duty-cycled sensor networks, and proposed a multi-channel based efficient broadcast protocol, which can achieve low delay and high delivery rate. In [21], the authors proposed a novel collision-tolerant broadcast scheduling strategy for duty-cycled sensor networks, this strategy provides the chance to further reduce broadcasting delay by allowing collisions at non-critical nodes to speed up the broadcast process for critical ones. In [22], the authors proposed an energy-efficient broadcast redundancy minimization scheduling scheme for low-duty-cycle sensor networks. It first finds a set of forwarders that minimizes the number of broadcast transmissions, then constructs a forest of sub-trees based on the relationship between each forwarders and its corresponding receivers, a broadcast tree is ultimately constructed by connecting all sub-trees with a minimum number of connectors. Jiang et al. [23] came up with a probabilistic and fully-distributed broadcasting mechanism, which has been verified the high efficiency of delay performance and energy performance. In [24], the authors investigated the energy efficient broadcast problem with minimum delay constraint for low-duty-cycle WSNs with unreliable links, and proposed a distributed heuristic solution. Cheng et al. [25] proposed a novel dynamic switching based reliable flooding framework, which is designed as an enhancement layer to provide efficient and reliable delivery for a variety of existing flooding tree structures in low-duty-cycle WSNs. In [26], the authors focused on the delay-constrained minimum energy broadcast problem for low-duty-cycle WSNs, they proposed an efficient heuristic solution and verified its high efficiency by extensive simulations [27]. first verified the existence of link correlation in sensor networks with unreliable links by extensive experiments, then utilized such link correlation to come up with a novel energy-efficient broadcast scheme for low-duty-cycle WSNs. The key idea is to make nodes with high correlation be associated with a common sender and have the identical working-sleeping schedule. In [32], the authors studied the minimum-transmission broadcast problem for duty-cycled WSNs and presented a novel level-based approximation scheme to build a broadcast backbone in order to minimize the total number of transmissions, they devised an approximation algorithm to construct the broadcast backbone, which provides a near optimal solution for the target problem [34]. considered how to fully utilize the broadcast nature of wireless media to solve the minimum energy broadcast problem with minimum delay constraint for low-duty-cycle WSNs. However, [34] assumed a strict broadcasting transmission model where each forwarder cannot send the beacon packets until the broadcasting message is received, this implies the improvement of energy efficiency will be limited.

Currently, very a few the existing works considered the generalized broadcasting performance, which consists of both delay performance and energy performance, as the optimization objective. The typical related work is [39]. In [39], the authors investigated the broadcast problem for duty-cycle-aware WSNs, they first defined a cost function which characterizes the tradeoff between the broadcasting delay and the total energy consumption for broadcasting. By assigning different tradeoff parameters in the cost function, the performance requirements from various applications can be covered. Then, an efficient broadcasting schedule was presented to minimize the cost function. The broadcasting cost function proposed in this paper is generally similar to that proposed in [39]. They both provide a flexible control on the tradeoff between broadcasting delay and total energy consumption for broadcasting to characterize the generalized broadcasting performance. However, it has some differences between their detailed definitions. In [39], specifically, it utilizes the coverage broadcasting delay (*i.e.*, the maximum of E2E broadcasting delay) to characterize broadcasting delay performance, and employs two parameters to represent the tradeoff factor. Different from the literature [39], the broadcasting cost function proposed in this paper utilizes the average E2E broadcasting delay to characterize broadcasting delay performance, and employs only one parameter to represent the tradeoff factor. Note that, [39] adopted the inefficient traditional broadcasting transmission model, in which every node will receive the broadcasting message at its scheduled wake-up time and the inherent broadcast nature of wireless media is totally ignored. Also, the work in [39] is not applicable for the case where the duty cycle is so low that the neighboring nodes almost have totally different working-sleeping schedules. Our recent work [14] considered how to fully exploit the broadcast nature of wireless media to optimize the generalized broadcasting performance for low-duty-cycle WSNs. In [14], we improved the broadcasting transmission model made in [34] by allowing a node to send beacon packets to its next-hop nodes immediately after its reception on a beacon packet, and proposed a top-down solution. Different from the work in [14], in this paper, we devise a novel and efficient bottom-up algorithm, which greatly outperforms the top-down solutions, to address our target problem.

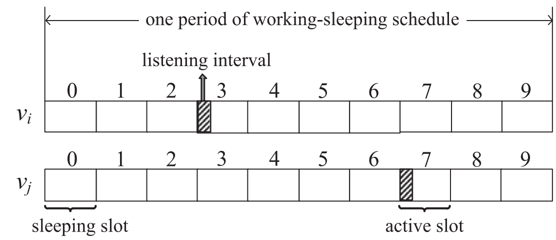


Fig. 1. The periodic working-sleeping schedule with $L = 10,\ t(v_i) = 3,\ t(v_j) = 7$.

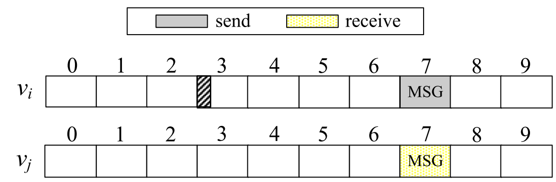


Fig. 2. Illustration of message transmission from v_i to v_j.

3. System model and problem statement

3.1. Network model and assumptions

In this paper, it is assumed that all nodes are uniformly deployed in a square sensory field where the sink is located at the center, and have an identical communication range. Here, we divide time into a number of equal time slots and regard each time slot as a basic unit of time. For simplicity and without loss of generality, we assume that every time slot is set long enough so that it can accommodate the transmission of any potential broadcasting message. The practical issue about the length of time slot will be then discussed in Section 5. Also, we assume the network works with low-duty-cycle mode, in which each node independently determines its own working-sleeping schedule. For simplicity and without loss of generality, the working-sleeping schedule of each node is assumed periodic and the period length is denoted by L. Specifically, each period of the working-sleeping schedule consists of one *active slot* and $L-1$ *sleeping slots*. In each *sleeping slot*, each node will turn off its radio but set a timer to wake up itself later. At the beginning of each *active slot*, a node v_i will wake itself up for a short duration of *listening interval*, to sense and listen to the channel. If any packet with the target receiver ID v_i is received, it will keep receiving until all packets of the message are received and then go to sleep immediately; otherwise, it will go to sleep immediately. Besides, if any sender is ready to communicate with any receiver, it will set a timer to wake up itself at the receiver's next *active slot*, then go to sleep immediately after the transmission. In other words, any node will not always keep awake during its whole *active slot*, and its awake duration depends on the size of the message transmitted in this *active slot*.

We denote by $t(v_i)$ the index of the *active slot* in each period of the working-sleeping schedule for any node v_i, where $0 \le t(v_i) \le L-1$. Fig. 1 explicitly illustrates an example of the periodic working-sleeping schedules for node v_i and node v_j, where $L = 10,\ t(v_i) = 3$ and $t(v_j) = 7$. In this example, the time slots in each working-sleeping schedule period are indexed by 0 to 9 in sequence, node v_i and node v_j will wake up at time slot 3 and time slot 7 of each working-sleeping schedule period, respectively. Fig. 2

illustrates the message transmission between duty-cycled nodes, where the sender v_i will set a timer to wake up itself to forward the message at time slot 7, *i.e.*, the *active slot* of the receiver v_j. Upon receiving the message, the receiver v_j will go to sleep immediately.

Let the undirected graph $G = (V, E)$ denote the network topology, where V represents the set of all nodes including one sink node v_0 and *Num* sensing nodes $\{v_1, \ldots, v_{Num}\}$, E represents the set of all undirected communication links. For any edge $(v_i, v_j) \in E$, we use $d(v_i, v_j)$ to denote the point-to-point transmission delay from node v_i to node v_j, which can be calculated as follows:

$$d(v_i, v_j) = \begin{cases} t(v_j) - t(v_i), & if\ t(v_j) > t(v_i); \\ t(v_j) - t(v_i) + L, & otherwise. \end{cases} \quad (1)$$

Specially, we define $d(v_i, v_i) = 0$ for any node $v_i \in V$.

As the same with many existing works, we also make the following basic assumptions: (1) Local time synchronization is achieved, which can be realized by the existing time synchronization protocol with less overhead [40]. (2) Each node is aware of the working-sleeping schedule of all its 1-hop and 2-hop neighboring nodes, which can be realized by just exchanging local information with the neighbors twice initially after the network is deployed. (3) This paper is mainly aimed at low-duty-cycle networks, where the duty cycle of each node is usually 1% or below. For simplicity, we assume that the potential packet collision issue in our target low-duty-cycle networks can be approximately ignored, this is because the low-duty-cycle mode inherently reduces the collision probability to a great extent which has been verified in [39]. Even for the very rare case with few collisions, we can simply adopt the traditional CSMA/CA protocol to well solve the collisions. (4) We assume the working-sleeping schedules of any node and its 1-hop neighbors are totally different from each other, which is usually true for real applications and can be implemented by adopting some existing power management protocols, *e.g.*, [41]. In order to improve the network performance, *e.g.*, to minimize average detection delay, the existing work [41] carefully designed the working schedules of all nodes in a fully distributed way to make the neighboring nodes rotate the sensory coverage. Actually, even for the rare special case where very a few neighboring nodes could wake up at the same time slot, our proposed solution in this paper can be still applicable, which will be discussed in Section 5. (5) We mainly consider the target networks with reliable links, namely the link qualities are all assumed 100%. For simplicity and without loss of generality, this paper does not consider the ACK packets, which usually cost rather less energy consumption. Further, we will discuss the ACK packets in Section 5.

3.2. Broadcasting transmission model

As we know, low-duty-cycle mode will make the wireless media lose its inherent advantage for broadcasting. Traditional broadcasting transmission model for low-duty-cycle WSNs will lead to energy-inefficient broadcasting, due to the fact that any local single-hop broadcast is implemented by multiple unicasts. In this paper, we adopt the novel opportunistic broadcasting transmission model, in which all receivers are divided into two types: *DelayedReceivers* and *InstantReceivers*. The basic idea is to allow any *DelayedReceiver* to postpone its wake-up time to opportunistically overhear the broadcasting message which is sent to some *InstantReceiver*. Specifically, all *InstantReceivers* will receive the broadcasting message only at their scheduled *active slot*, but, any *DelayedReceiver* will only receive a short beacon packet at its scheduled *active slot*. Upon receiving the short beacon packet, the *DelayedReceiver* will go to sleep immediately and set a timer to wake itself up at the next *active slot* of some *InstantReceiver*, of which ID is included in the beacon packet, to opportunistically overhear the broadcasting message. Note that, the beacon packets are not costly in our model, since the beacon packets are only the short control packets without any payload.

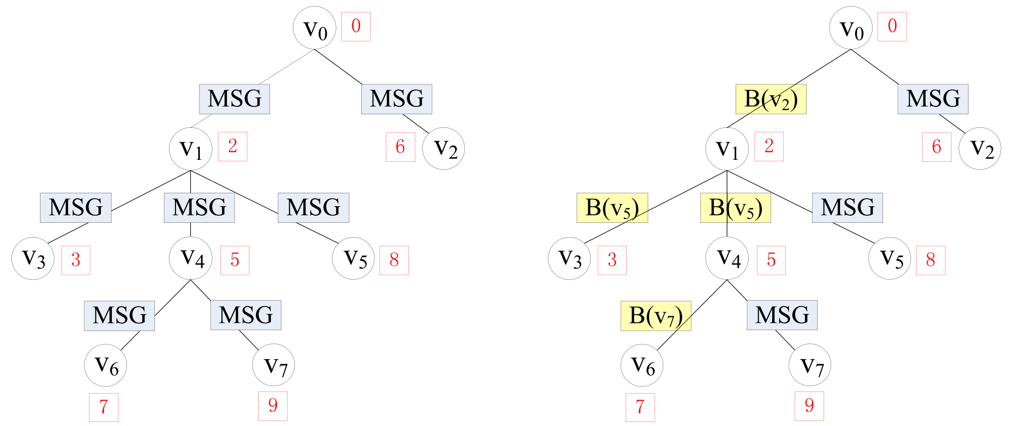
For multi-hop nodes, this model adopts an efficient pre-beacon scheme, that is, to allow any node to send beacon packets to its next-hop nodes immediately after its reception on a beacon packet. In other words, we allow any forwarder, who is the *DelayedReceiver*, to send the beacon packets to its next-hop neighbors between its beacon reception time and message reception time. By carefully designing the working schedules of all nodes in a multi-hop network, both the beacon packets and the broadcasting message can be transmitted in a timely way.

Fig. 3 illustrates a simple broadcast example on a tree topology, where the number labeled within each square denotes the scheduled *active slot* and L=10. Fig. 3(a) shows the case where the traditional broadcasting transmission model is adopted, all nodes will receive the broadcasting message at their scheduled *active slot*, the network will thus have the average broadcasting delay of (2+6+3+5+8+7+9)/7=5.7 and the total energy consumption of $E_a = 7 \times k \times e_s^d + 7 \times k \times e_r^d$, where k is a positive number and denotes the number of data packets in a broadcasting message, e_s^d and e_r^d denote the energy consumption of sending and receiving a data packet, respectively. Fig. 3(b) exhibits the solution with the opportunistic broadcasting transmission model. As shown in Fig. 3(b), the sink v_0 delivers the beacon packet $B(v_2)$, which only contains the ID of the *InstantReceiver* v_2, to the *DelayedReceiver* v_1, and then delivers the broadcasting message to the *InstantReceiver* v_2. Upon receiving $B(v_2)$, node v_1 will go to sleep immediately, and set a timer to wake itself up at the next scheduled *active slot* of the *InstantReceiver* v_2, *i.e.*, time slot 6, to opportunistically overhear the broadcasting message. Note that, since each node can be aware of the working-sleeping schedules of all its neighboring nodes within two hops, the *DelayedReceiver* v_1 can get the working-sleeping schedule of the *InstantReceiver* v_2. Upon receiving the beacon packet $B(v_2)$, the forwarder v_1 will respectively send a beacon packet $B(v_5)$ to the *DelayedReceivers* v_3 and v_4 at their scheduled *active slots, i.e.*, time slot 3 and time slot 5, such that their message reception time can be deferred to the scheduled *active slot* of the *InstantReceiver* v_5, *i.e.*, time slot 8. Likewise, the forwarder v_4 will send the beacon packet $B(v_7)$ to the *DelayedReceiver* v_6 immediately after it receives $B(v_5)$, to make v_6 overhear the broadcasting message which is sent from the forwarder v_4 to the *InstantReceiver* v_7. The above broadcast process is explicitly illustrated by Fig. 4. We can find that v_1 and v_2 will have the same E2E broadcasting delay 6, v_3, v_4 and v_5 will have the same E2E broadcasting delay 8, v_6 and v_7 will have the same E2E broadcasting delay 9. Thus, the solution in Fig. 3(b) will have the average broadcasting delay of (6+6+8+8+8+9+9)/7=7.7 and the total energy consumption of $E_b = 4 \times (e_s^b + e_r^b) + 3 \times k \times e_s^d + 7 \times k \times e_r^d$, where e_s^b and e_r^b denote the energy consumption of sending and receiving a beacon packet, respectively.

Compared with the traditional solution in Fig. 3(a), the total energy benefit of the solution in Fig. 3(b) will be $E_\Delta = E_a - E_b = 4 \times (k \times e_s^d - (e_s^b + e_r^b))$. As shown in [42], it is usual that a data packet has a length of 133 bytes and a beacon packet has only a length of 19 bytes, $e_s^b + e_r^b$ is thus far less than e_s^d in practice, which means E_Δ must be greater than 0. For broadcasting applications with large messages, *e.g., code update*, the benefit E_Δ will be significant as $k \gg 1$. In other words, the opportunistic broadcasting transmission model can lead to less total energy consumption but higher average broadcasting delay than the traditional one.

Further, we can have the following observation.

Observation 1. Given a solution with the opportunistic broadcasting transmission model, which contains K DelayedReceivers, its to-



(a) Broadcast without deferring (b) Broadcast with 4 *DelayedReceivers*

Fig. 3. An example with the opportunistic broadcasting transmission model.

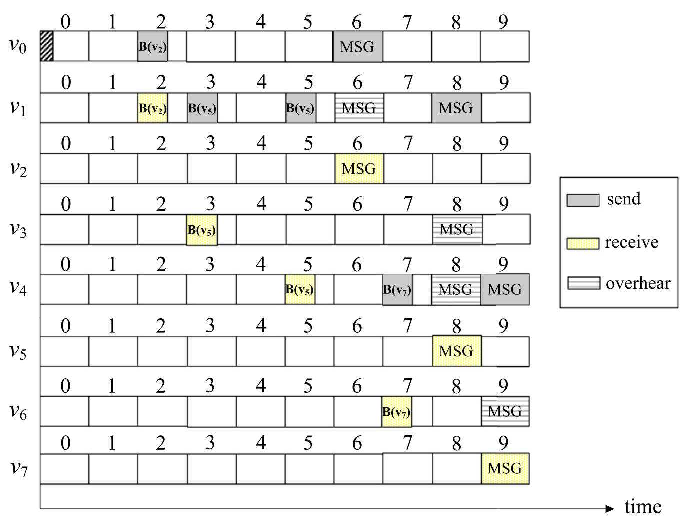


Fig. 4. Illustration of the solution in Fig. 3(b).

tal energy benefit compared with the traditional solution will be $E_{\Delta} = K \times (k \times e_s^d - (e_s^b + e_r^b))$.

According to the above observation, we find that more *DelayedReceivers* will yield less total energy consumption for broadcasting, however, could result in higher average broadcasting delay, which implies the opportunistic broadcasting transmission model can provide a flexible control on the tradeoff between average broadcasting delay and total energy consumption for broadcasting. For the solutions with the opportunistic broadcasting transmission model, thus, the total energy consumption for broadcasting can be essentially characterized by the number of the *InstantReceivers, i.e.*, the solution with fewer number of the *InstantReceivers* will have less total energy consumption for broadcasting.

3.3. Problem statement

In this paper, our optimization objective is the generalized broadcasting performance that can adaptively characterize any tradeoff relationship between delay performance and energy performance. Given any network topology $G = (V, E)$, we can define the following broadcasting cost function:

$$COST(G) = DPI + \eta \times EPI, \tag{2}$$

where DPI denotes delay performance index that characterizes the delay performance, EPI denotes energy performance index that characterizes the energy performance, and the non-negative parameter η denotes a tradeoff factor. We can find that the generalized broadcasting performance can be well characterized by the above broadcasting cost function $COST(G)$. By adaptively adjusting the tradeoff factor η, our optimization objective $COST(G)$ can be universally applicable for the applications with various broadcasting performance requirements.

Here, we take the average broadcasting delay, which denotes the average of end-to-end (E2E) delay from the sink to all nodes in the network and is usually an important metric to evaluate broadcasting performance, to characterize DPI. Let $D(v_i)$ and $D^*(v_i)$ respectively denote the real E2E broadcasting delay and the theoretically optimal E2E broadcasting delay from the sink node to any sensor node v_i. Given any network topology $G = (V, E)$, the average broadcasting delay can be represented as follows:

$$\widehat{D}(G) = \frac{\sum_{v_i \in V} D(v_i)}{|V|} = \frac{\sum_{v_i \in V} D^*(v_i) + \sum_{v_i \in V} (D(v_i) - D^*(v_i))}{|V|} \tag{3}$$

where $|V|$ denotes the total number of nodes in G, and if v_i is the sink node, then $D(v_i) = D^*(v_i) = 0$.

For any broadcasting schedule on G, both $\sum_{v_i \in V} D^*(v_i)$ and $|V|$ in Eq. (3) must be fixed. Accordingly, the average broadcasting delay can be essentially characterized by the following equation:

$$\Delta_{delay} = \sum_{v_i \in V} (D(v_i) - D^*(v_i)), \tag{4}$$

where $D(v_i) - D^*(v_i)$ denotes the increased E2E broadcasting delay compared with the theoretically optimum for any node v_i. In this paper, we will simply employ Δ_{delay} to characterize DPI. For the broadcasting schedules based on the opportunistic broadcasting transmission model, as stated in Section 3.2, the total energy consumption can be essentially characterized by the number of the *InstantReceivers*. Thus, we can simply employ the number of the *InstantReceivers, i.e.*, the number of the broadcasting message transmissions, to characterize EPI.

In this paper, our objective is to solve the following *Minimum Cost Broadcast Problem* (MCB).

Problem 1 (MCB)**.** Given any low duty cycle sensor network $G = (V, E)$, how to utilize the opportunistic broadcasting transmission model to design an efficient broadcasting schedule M, to minimize the following broadcasting cost function:

$$COST(G) = \Delta_{delay} + \eta \times C(M) \tag{5}$$

where $C(M)$ denotes the number of InstantReceivers in M, and the parameter η denotes a non-negative tradeoff factor that is usually application-specific in practice.

4. Solution

In this section, we first define a simplified problem, *i.e., Receiver-Constrained Minimum Cost Single-hop Broadcast Problem* (RC-MCSB), and propose an optimal solution. Then, we further extend the solution of RC-MCSB problem to our target problem and come up with an efficient bottom-up algorithm.

4.1. Receiver-constrained minimum cost single-hop broadcast

Let $G_s(\{v_i\}\bigcup\{v_i^1,\ldots,v_i^N\},E_s)$ denote a single-hop low-duty-cycle network topology that consists of one sender v_i and N receivers $\{v_i^1,\ \ldots,\ v_i^N\}$. Here, the receivers $\{v_i^1,\ \ldots,\ v_i^N\}$ are sorted according to the ascending order of the sleep latency from the sender to them, namely $d(v_i,\ v_i^j) < d(v_i,\ v_i^{j+1})$ $(1 \le j \le N-1)$, and E_s is the set of the edges $\{(v_i,v_i^1),(v_i,v_i^2),\ldots,(v_i,v_i^N)\}$.

Definition 1 (Constrained Range)**.** Given a single-hop low-duty-cycle network topology $G_s(\{v_i\}\bigcup\{v_i^1,\ldots,v_i^N\},E_s)$ and $d(v_i,\ v_i^j) < d(v_i,\ v_i^{j+1})$ $(1 \le j \le N-1)$, $CR(v_i^j)$ is called the constrained range of a receiver $v_i^j(1 \le j \le N)$ if and only if

$$CR(v_i^j) = \left\{v_i^j, v_i^{j+1}, \ldots, v_i^{n_j}\right\} \tag{6}$$

where n_j is a integer parameter and $j \le n_j \le N$.

For example, if N=5 and j=3, then the feasible constrained range of v_i^3 could be $\{v_i^3\}$, $\{v_i^3,v_i^4\}$ and $\{v_i^3,v_i^4,v_i^5\}$.

Here, we first consider a simple variant of our target problem, namely the following *Receiver-Constrained Minimum Cost Single-hop Broadcast Problem* (RC-MCSB).

Problem 2 (RC-MCSB)**.** Given a single-hop low-duty-cycle network $G_s(\{v_i\}\bigcup\{v_i^1,\ldots,v_i^N\},E_s)$, if the constrained range of each receiver is given, how to design an efficient broadcasting schedule M, so that the broadcasting cost function in Eq. (5) is minimized, which is subject to the constraint that each receiver's constrained range should contain at least one InstantReceiver.

For a single-hop network topology, the broadcasting schedule mainly depends on the forwarding decision of the sender. Specifically, the design of the broadcasting schedule in any single-hop topology is essentially to determine which receivers are the *DelayedReceivers* and the *InstantReceivers*, and to indicate the corresponding *InstantReceiver* of each *DelayedReceiver*. Note that, if the receiver v_i^k is the corresponding *InstantReceiver* of the *DelayedReceiver* v_i^j, it means that the sender will send a beacon packet $B(v_i^k)$ to the *DelayedReceiver* v_i^j and then forward the broadcasting message to the *InstantReceiver* v_i^k. Upon receiving $B(v_i^k)$, the *DelayedReceiver* v_i^j will go to sleep immediately, and set a timer to wake itself up at the next *active slot* of the *InstantReceiver* v_i^k to opportunistically overhear the broadcasting message. For better description, here, we employ $IR(v_i^j)$ to denote the corresponding *InstantReceiver* of any *DelayedReceiver* v_i^j. We can get the following observation.

Observation 2. Given a single-hop low-duty-cycle network $G_s(\{v_i\}\bigcup\{v_i^1,\ldots,v_i^N\},E_s)$, if nodes in any receiver-subset $\{v_i^j,\ldots,v_i^k\}$ $(1 \le j \le k \le N-1)$ are all determined as DelayedReceivers and node v_i^{k+1} is determined as the InstantReceiver, then the corresponding InstantReceiver of any node in $\{v_i^j,\ldots,v_i^k\}$ must be v_i^{k+1}, i.e., $IR(v_i^t) = v_i^{k+1}$ $(j \le t \le k)$.

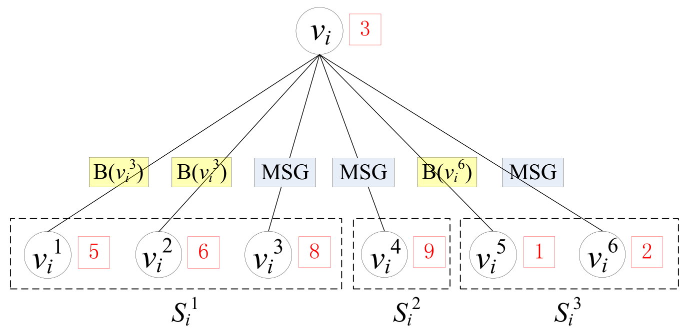


Fig. 5. An example of a partition.

Observation 2 is true due to the fact that if the corresponding *InstantReceiver* of any *DelayedReceiver* v_i^t $(j \le t \le k)$ is set as not v_i^{k+1} but any *InstantReceiver* in $\{v_i^{k+2},\ldots,v_i^N\}$, it will not affect the number of the *InstantReceivers* in broadcasting schedule but will result in longer Δ_{delay}. According to Observation 2, we can present the definition of *coverage group* as follows:

Definition 2 (Coverage Group)**.** Given a single-hop low-duty-cycle network $G_s(\{v_i\}\bigcup\{v_i^1,\ldots,v_i^N\},E_s)$, a coverage group is defined as a receiver-subset $\{v_i^j,\ldots,v_i^k\}$ $(1 \le j \le k \le N)$, where only the receiver v_i^k is set as the InstantReceiver and any receiver v_i^t $(j \le t \le k-1)$ is set as the DelayedReceiver with $IR(v_i^t) = v_i^k$.

Essentially, a *coverage group* can explicitly characterize a part of the sender's forwarding decision. Specifically, a *coverage group* $\{v_i^j,\ldots,v_i^k\}$ $(1 \le j < k \le N)$ implies the forwarding decision that the sender will respectively send the beacon packet $B(v_i^k)$ to each *DelayedReceiver* v_i^t $(j \le t \le k-1)$ and then forward the broadcasting message to the *InstantReceiver* v_i^k. Specially, $j = k$ in the *coverage group* implies the forwarding decision that the sender will only forward the broadcasting message to the *InstantReceiver* v_i^k. In other words, any *coverage group* represents the set of receivers covered by one broadcasting message transmission.

Given any *coverage group* $S = \{v_i^j,\ldots,v_i^k\}$ $(1 \le j \le k \le N)$, we assume that *groupIR*(S) denotes the ID of the *InstantReceiver* in *coverage group S;* $d_{sum}(S)$ denotes the sum of the increased E2E broadcasting delay compared with the theoretically optimum for all receivers in *coverage group S*; and *cost*(S) denotes the broadcasting cost resulted from *coverage group S*. Specifically,

$$groupIR(S) = v_i^k;$$

$$d_{sum}(S) = \sum_{t=j}^{k}\left(D(v_i^t) - D^*(v_i^t)\right) = \sum_{t=j}^{k} d\left(v_i^t, v_i^k\right);$$

$$cost(S) = d_{sum}(S) + \eta = \sum_{t=j}^{k} d\left(v_i^t, v_i^k\right) + \eta. \tag{7}$$

Note that, there is only one *InstantReceiver* in *coverage group S*. According to the definition of broadcasting cost, *i.e.*, Eq. (5), the second item in *cost*(S) must be η.

Definition 3 (Partition)**.** Given a single-hop low-duty-cycle network $G_s(\{v_i\}\ \bigcup\{v_i^1,\ \ldots,\ v_i^N\},\ E_s)$, a set of coverage groups $\{S_i^1,S_i^2,\ldots,S_i^M\}$ $(1 \le M \le N)$ is called a partition of receivers $\{v_i^1,\ldots,v_i^N\}$ if and only if (1) $\bigcup_{t=1}^{M} S_i^t = \{v_i^1,\ldots,v_i^N\}$; (2) $S_i^j \bigcap S_i^k = \emptyset$ for any two coverage groups S_i^j and S_i^k $(1 \le j < k \le M)$.

For a single-hop low-duty-cycle network, any forwarding decision of the sender can be well characterized by a partition of receivers. Fig. 5 illustrates an example of a partition $P_i = \{S_i^1,S_i^2,S_i^3\}$ where the coverage group $S_i^1 = \{v_i^1,v_i^2,v_i^3\}$, $S_i^2 = \{v_i^4\}$ and $S_i^3 =$

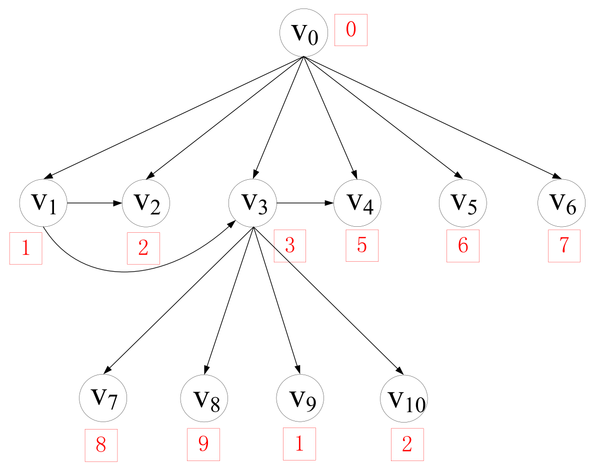


Fig. 6. Illustration of feasible/infeasible candidate child (L=10).

$\{v_i^5, v_i^6\}$. In this example, P_i essentially represents the broadcasting schedule that $\{v_i^3, v_i^4, v_i^6\}$ are the *InstantReceivers* and $\{v_i^1, v_i^2, v_i^5\}$ are the *DelayedReceivers* with $IR(v_i^1) = IR(v_i^2) = v_i^3$, $IR(v_i^5) = v_i^6$. Accordingly, the RC-MCSB problem is actually equivalent to the following optimal partition problem.

Problem 3 (Optimal Partition Problem)**.** Given a single-hop low-duty-cycle network $G_s(\{v_i\} \bigcup \{v_i^1, \ldots, v_i^N\}, E_s)$ and the constrained range $CR(v_i^j)$ for each receiver v_i^j $(1 \le j \le N)$, how to find an efficient partition P_i of receivers $\{v_i^1, \ldots, v_i^N\}$, to achieve the following optimization of the broadcasting cost:

$$\min \sum_{S \in P_i} cost(S) \tag{8}$$

which is subject to the constraints that for each receiver v_i^j $(1 \le j \le N)$, there is at least one coverage group $S \in P_i$ such that $groupIR(S) \in CR(v_i^j)$.

Here, we let $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ $(1 \le k \le N)$ denote the following problem: *Given a single-hop low-duty-cycle network* $G_s(\{v_i\} \bigcup \{v_i^1, \ldots, v_i^N\}, E_s)$ *and the constrained range* $CR(v_i^j)$ *for each receiver* v_i^j $(1 \le j \le N)$*, suppose that* $\{v_i^1, \ldots, v_i^k\}$ *is the subset of receivers* $\{v_i^1, \ldots, v_i^N\}$ *where* $1 \le k \le N$*, how to find an efficient partition* P_i *of receivers* $\{v_i^1, \ldots, v_i^k\}$*, to minimize the broadcasting cost* $\sum_{S \in P_i} cost(S)$*, which is subject to the constraints that for each receiver* v_i^j $(1 \le j \le k)$*, there is at least one coverage group* $S \in P_i$ *such that* $groupIR(S) \in CR(v_i^j) \bigcap \{v_i^j, \ldots, v_i^k\}$.

For any receiver v_i^t $(1 \le t \le k)$, 1) if $CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^k\} = \{v_i^t, \ldots, v_i^k\}$, we will call $CR(v_i^t)$ an *invalid constrained range* in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$. This is because for any feasible solution to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$, v_i^k must be the *InstantReceiver* in the last *coverage group* S_i^M, which implies the constraint on v_i^t must be satisfied, *i.e.*, it must have $groupIR(S_i^M) \in CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^k\}$; 2) if $CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^{k-1}\} = CR(v_i^t)$, then we will call $CR(v_i^t)$ a *valid constrained range* in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$.

Further, we denote by $OPT(k)$ the optimal broadcasting cost for the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$. When $k = N$, $CR(v_i^j) \bigcap \{v_i^j, \ldots, v_i^N\} = CR(v_i^j)$ for any receiver v_i^j $(1 \le j \le N)$, which implies the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^N\})$ will be equal to the optimal partition problem. Therefore, our target is to get $OPT(N)$ and the optimal solution to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^N\})$.

Lemma 1. *Given a problem* $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ *and its any subproblem* $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$ $(1 \le j < k)$*, if* P_i *is a feasible solution to the subproblem* $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$*, it must exist a feasible solution* P_i' *to the problem* $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$*, such that* P_i *is a part of* P_i'*, i.e.,* $P_i \subset P_i'$.

Proof. For any receiver v_i^t $(1 \le t \le j)$, if $CR(v_i^t)$ is a valid constrained range in the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$, then $CR(v_i^t)$ must also be a valid constrained range in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$, and

$$CR\big(v_i^t\big) \bigcap \big\{v_i^t, \ldots, v_i^j\big\} = CR\big(v_i^t\big) \bigcap \big\{v_i^t, \ldots, v_i^k\big\} = CR\big(v_i^t\big);$$

if $CR(v_i^t)$ is an invalid constrained range in the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$, then it must have

$$CR\big(v_i^t\big) \bigcap \big\{v_i^t, \ldots, v_i^j\big\} \subseteq CR\big(v_i^t\big) \bigcap \big\{v_i^t, \ldots, v_i^k\big\}.$$

We suppose that P_i is a feasible solution to the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$. For each receiver v_i^t $(1 \le t \le j)$, it must exist a coverage group $S \in P_i$ such that $groupIR(S) \in CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^j\} \subseteq CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^k\}$, which implies the feasible solution P_i also satisfies the constraint on each v_i^t $(1 \le t \le j)$ in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$. Thus, P_i must be a part of some feasible solution to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$. The proof is completed. □

Theorem 1. *The solution to the problem* $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ *has the property of optimal substructure.*

Proof. Suppose that $P_i^* = \{S_i^1, \ldots, S_i^M\}$ is the optimal solution to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$, the partition $\{S_i^1, \ldots, S_i^K\}$ $(1 \le K < M)$, which is a part of P_i^*, must be a feasible solution to the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$ where $v_i^j = groupIR(S_i^K)$ and $j < k$. This is because for any receiver v_i^t $(1 \le t \le j)$, 1) if $CR(v_i^t)$ is a valid constrained range in the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$, then $CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^j\} = CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^k\}$, which implies the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ and the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$ have the same constraint on v_i^t, that is, the solution $\{S_i^1, \ldots, S_i^K\}$, which is the subset of P_i^*, must satisfy the constraint on v_i^t in the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$; 2) if $CR(v_i^t)$ is an invalid constrained range in the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$, then it must have $v_i^j \in CR(v_i^t) \bigcap \{v_i^t, \ldots, v_i^j\}$. Since $v_i^j = groupIR(S_i^K)$, the solution $\{S_i^1, \ldots, S_i^K\}$ must satisfy the constraint on v_i^t in the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$.

We assume that the feasible solution $\{S_i^1, \ldots, S_i^K\}$ to the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$ is not optimal, then it must exist a better feasible solution P_i^Δ to the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$. According to Lemma 1, P_i^Δ must satisfy the constraint on each v_i^t $(1 \le t \le j)$ in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$. This implies the partition $P_i^\Delta \bigcup \{S_i^{K+1}, \ldots, S_i^M\}$ must be a better feasible solution than $P_i^* = \{S_i^1, \ldots, S_i^K\} \bigcup \{S_i^{K+1}, \ldots, S_i^M\}$ to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$, which conflicts with the assumption that P_i^* is the optimal solution to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$. Thus, $\{S_i^1, \ldots, S_i^K\}$, which is a part of the optimal solution P_i^* to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$, must be the optimal solution to the subproblem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^j\})$, the proof is completed. □

According to Theorem 1, we can thus adopt a dynamic programming approach to address the optimal partition problem. For better description, here, we define a virtual receiver v_i^0 with $CR(v_i^0) = \emptyset$.

Theorem 2. *$OPT(k)$ has the following recurrence:*

$$OPT(k) = \min_{s(k) \le j \le k-1} \left\{ OPT(j) + cost\left(\left\{v_i^{j+1}, \ldots, v_i^k\right\}\right)\right\} \tag{9}$$

where $OPT(0) = 0$, $cost(\{v_i^{j+1}, \ldots, v_i^k\}) = \sum_{t=j+1}^{k} d(v_i^t, v_i^k) + \eta$, and $s(k) = \max\{s \in \{0, \ldots, k-1\} | v_i^k \notin CR(v_i^s)\}$.

Proof. Suppose that $\{S_i^1, \ldots, S_i^M\}$ is the optimal solution to the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$, v_i^k must be the *InstantReceiver* in the last coverage group S_i^M. If we assume v_i^j is the *InstantReceiver* in the coverage group S_i^{M-1}, it must have $OPT(k) = OPT(j) + cost(\{v_i^{j+1}, \ldots, v_i^k\})$ due to the property of optimal substructure. If there is no constrained range for each receiver, the range of j must be from 0 to $k-1$. For the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ where each receiver has a given constrained range, the range of j must be within a limited range, namely

$$OPT(k) = \min_{j \in R} \left\{ OPT(j) + cost\left(\left\{v_i^{j+1}, \ldots, v_i^k\right\}\right)\right\} \tag{10}$$

where the limited range $R \subseteq \{0, \ldots, k-1\}$.

For any receiver v_i^j $(1 \le j \le k-1)$, if $CR(v_i^j)$ is an invalid constrained range in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$, the constraint on v_i^j in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ must be satisfied since v_i^k must be the *InstantReceiver*. Thus, we only need to consider the constraints on receivers with valid constrained ranges. Let $v_i^{s(k)}$ denote the last receiver with the valid constrained range in $\{v_i^0, v_i^1, \ldots, v_i^{k-1}\}$, namely $s(k) = \max\{s \in \{0, \ldots, k-1\} | v_i^k \notin CR(v_i^s)\}$.

To determine the range of R, we make a discussion by dividing $\{0, \ldots, k-1\}$ into the following three intervals:

(1) If $0 \le j < s(k)$, then it must have $v_i^{s(k)} \in S_i^M$ due to the fact that v_i^j is the *InstantReceiver* in S_i^{M-1}. Since $v_i^k \notin CR(v_i^{s(k)})$, all receivers in $CR(v_i^{s(k)})$ must be the *DelayedReceivers* in the coverage group S_i^M, which implies the constraint on $v_i^{s(k)}$ in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ must not be satisfied. So, this interval is not feasible.

(2) If $s(k) \le j < s(k) + |CR(v_i^{s(k)})|$, there is at least one *InstantReceiver* in $CR(v_i^{s(k)})$ since v_i^j is the *InstantReceiver*, which implies the constraint on $v_i^{s(k)}$ in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$ must be satisfied. So, this interval is feasible.

(3) If $s(k) + |CR(v_i^{s(k)})| \le j \le k-1$, each $CR(v_i^t)$ $(j+1 \le t \le k)$ must not be a valid constrained range in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^k\})$. So, this interval is also feasible.

Overall, the range of R is $\{s(k), \ldots, k-1\}$. The proof is thus completed. □

Algorithm 1 explicitly shows how to solve the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^N\})$ based on the recurrence in Theorem 2. In Line 3 to Line 12, we first in turn compute $OPT[1], OPT[2], \ldots, OPT[N]$ according to the recurrence (9). Here, $lastIR[k] = j$ implies $OPT[k]$ is resulted from $OPT[j] + cost(\{v_i^{j+1}, \ldots, v_i^k\})$, in other words, if v_i^k is the *InstantReceiver* in the optimal solution, then $v_i^{lastIR[k]}$ must be the last *InstantReceiver* of v_i^k in the optimal solution. As shown in Line 13 to Line 26, we will get the optimal partition by a iterative way. As v_i^N must be the *InstantReceiver* in the optimal solution, we can initially set $k = N$, and define $S[1]$ as the last *coverage group* in the optimal solution, we must have $S[1] = \{v_i^{lastIR[k]+1}, \ldots, v_i^k\}$, and $IR(v_i^j) = v_i^k$ for all $lastIR[k] + 1 \le j \le k-1$. Next, we set $k = lastIR[k]$ and v_i^k must be the *InstantReceiver* in $S[2]$, then we repeat the above iteration process to figure out all the *coverage groups* $S[1], S[2], S[3], \ldots$ in the optimal solution P_i^*.

As shown in Algorithm 1, we can employ Eq. (7) to compute $cost(\{v_i^j, \ldots, v_i^k\})$ in $O(N)$ time for any coverage group $\{v_i^j, \ldots, v_i^k\}$.

Algorithm 1: Optimal Partition Algorithm.

Input: $G_s(\{v_i\} \bigcup \{v_i^1, \ldots, v_i^N\}, E_s)$ and $CR(v_i^j)$ $(1 \le j \le N)$.
Output: $OPT(N)$ and the optimal partition P_i^*.

1 $OPT[0] = 0$; $S[j] = \emptyset$ $(1 \le j \le N)$; //$S[j]$ denotes the coverage group.
2 compute all $cost(\{v_i^j, \ldots, v_i^k\})$ $(1 \le j \le k \le N)$ according to the Equation 7;
3 **for** $k = 1$ *to* N **do**
4     $minCost = +\infty$; $s_k = \max\{s \in \{0, \ldots, k-1\} | v_i^k \notin CR(v_i^s)\}$;
5   **for** $j = s_k$ *to* $k-1$ **do**
6   **if** $OPT[j] + cost(\{v_i^{j+1}, \ldots, v_i^k\}) < minCost$ **then**
7             $minCost = OPT[j] + cost(\{v_i^{j+1}, \ldots, v_i^k\})$;
8             $lastIR[k] = j$; //$OPT[k]$ is resulted from $OPT[j] + cost(\{v_i^{j+1}, \ldots, v_i^k\})$.
9   **end**
10   **end**
11     $OPT[k] = minCost$;
12 **end**
13 $groupCount = 0$; $k = N$;
14 **for** $j = N$ *to* 1 **do**
15   **if** $j == k$ **then**
16         $groupCount = groupCount + 1$;
17         $S[groupCount] = S[groupCount] \bigcup \{v_i^j\}$;
18   set v_i^j as the *InstantReceiver*;
19         $groupIR = k$; $k = lastIR[k]$;
20   **end**
21   **else**
22         $S[groupCount] = S[groupCount] \bigcup \{v_i^j\}$;
23   set v_i^j as the *DelayedReceiver* with $IR(v_i^j) = v_i^{groupIR}$;
24   **end**
25 **end**
26 $P_i^* = \{S[groupCount], S[groupCount-1], \ldots, S[1]\}$;

Since it has totally $O(N^2)$ coverage groups for receivers $\{v_i^1, \ldots, v_i^N\}$, the total running time to compute all $cost(\{v_i^j, \ldots, v_i^k\})(1 \le j \le k \le N)$ will be $O(N^3)$. We can find that the running time of Algorithm 1 is dominated by the time to compute the broadcasting costs of all the coverage groups, thus, Algorithm 1 has a time complexity of $O(N^3)$.

Specially, we can find that if each $CR(v_i^j)$ $(1 \le j \le N)$ is an invalid constrained range in the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^N\})$ (*i.e.*, $CR(v_i^j) = \{v_i^j, \ldots, v_i^N\}$), the problem $\hat{P}(v_i, \{v_i^1, \ldots, v_i^N\})$ will be equivalent to the *Minimum Cost Broadcast Problem* (MCB) for single-hop networks, and $s(k)$ will always be 0 for any $OPT(k)$.

4.2. The bottom-up algorithm

Next, we will introduce how to solve our target MCB problem based on the solution to the RC-MCSB problem. Specifically, we propose a novel bottom-up solution to the MCB problem, which mainly consists of the following three phases: (1) fat-tree construction; (2) bottom-up forwarder selection; (3) schedule adjustment.

4.2.1. Fat-tree construction

Given a low-duty-cycle sensor network $G = (V, E)$ and any node $v_i \in V$, we let $CPS(v_i)$ and $CCS(v_i)$ respectively denote node v_i's *candidate parents set* and *candidate children set*. Specifically, any node $v_j \in CPS(v_i)$ if and only if there exists a minimum end-to-end delay path from the sink to node v_i where node v_j is the parent of node v_i; Likewise, any node $v_j \in CCS(v_i)$ if and only if there exists a minimum end-to-end delay path from the sink to node v_j where node v_i is the parent of node v_j.

By adopting a simple approach that is similar to the classical *Dijkstra Algorithm*, we can get the *Minimum Delay Path Fat-Tree* (MDPFT) on G, where each node v_i can be aware of its $CPS(v_i)$ and $CCS(v_i)$. Specially, $CPS(v_0)$ is defined as $\emptyset$ for the sink v_0. Obviously, any spanning subtree of the MDPFT must be a *Minimum Delay Path Tree* (MDPT).

4.2.2. Bottom-up forwarder selection

Given a low-duty-cycle sensor networks $G = (V, E)$, we first present the definition of *feasible/infeasible candidate child* as follows.

Definition 4 (Feasible/Infeasible Candidate Child)**.** For any single-hop topology $G_s(\{v_i\} \bigcup CCS(v_i), E_s)$ where $v_i \in V$ and $CCS(v_i) \neq \emptyset$,

(1) any node $v_j \in CCS(v_i)$ is called the feasible candidate child of v_i if and only if $CCS(v_j) = \emptyset$ or $CCS(v_j) \subset CCS(v_i)$;

(2) any node $v_j \in CCS(v_i)$ is called the infeasible candidate child of v_i if and only if $CCS(v_j) \not\subset CCS(v_i)$.

Fig. 6 exhibits a simple network topology where $CCS(v_0) = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $CCS(v_1) = \{v_2, v_3\}$, $CCS(v_3) = \{v_4, v_7, v_8, v_9, v_{10}\}$ and $CCS(v_2) = CCS(v_4) = CCS(v_5) = CCS(v_6) = CCS(v_7) = CCS(v_8) = CCS(v_9) = CCS(v_{10}) = \emptyset$. According to the Definition 4, nodes $\{v_1, v_2, v_4, v_5, v_6\}$ must be the feasible candidate children of node v_0, and node v_3 must be the infeasible candidate child of node v_0.

Here, we will find a MDPT from the MDPFT to design the broadcasting schedule. Different from the traditional top-down solutions, we come up with a novel *Bottom-up Forwarder Selection Algorithm* to get the broadcasting schedule. For better description, we first define the following *Delay-Bounded Minimum Cost Single-hop Broadcast Problem* (DB-MCSB).

Problem 4 (DB-MCSB)**.** Given a single-hop low-duty-cycle network $G_s(\{v_i\} \bigcup \{v_i^1, \ldots, v_i^N\}, E_s)$ and the deferred delay bound $d^{\triangle}(v_i^j)$ for any receiver v_i^j $(1 \le j \le N)$, how to utilize the opportunistic broadcasting transmission model to design an efficient broadcasting schedule M so that the broadcasting cost function in Eq. (5) is minimized, which is subject to the constraint that each receiver v_i^j's deferred delay $D(v_i^j) - D^*(v_i^j)$ must be less than $d^{\triangle}(v_i^j)$.

Obviously, the DB-MCSB problem is equivalent to the RC-MCSB problem that proposed in Section 4.1. For any receiver v_i^j $(1 \le j \le N)$, specifically, the constraint that $D(v_i^j) - D^*(v_i^j) < d^{\triangle}(v_i^j)$ is actually equivalent to the constraint that the constrained range $CR(v_i^j) = \{v | v \in \{v_i^j, \ldots, v_i^N\}$ && $d(v_i^j, v) < d^{\triangle}(v_i^j)\}$ should contain at least one *InstantReceiver*. Thus, the DB-MCSB problem can be simply solved by Algorithm 2 .

Algorithm 2: Delay-Bounded Minimum Cost Single-hop Broadcast Algorithm.

Input: $G_s(\{v_i\} \bigcup \{v_i^1, \ldots, v_i^N\}, E_s)$ and $d^{\triangle}(v_i^j)$ $(1 \le j \le N)$.
Output: $OPT(N)$ and the optimal partition P_i^*.

```
for j = 1 to N do
    CR(v_i^j) = ∅;
    for k = j to N do
        if d(v_i^j, v_i^k) < d^△(v_i^j) then
            CR(v_i^j) = CR(v_i^j) ∪ {v_i^k};
        end
    end
end
call Algorithm 1;
```

Next, we will present the detailed description of the *Bottom-up Forwarder Selection Algorithm*, which is based on the solution to the DB-MCSB problem. Specifically, our proposed *Bottom-up Forwarder Selection Algorithm* consists of the following steps.

(1) *Step 1: Initialization*

First, we define two sets: *candidate forwarder set* and *determined forwarder set*, which are both initially set as $\emptyset$. For each node v_i where $v_i \in V$ and $CCS(v_i) \neq \emptyset$, it will check whether all nodes in $CCS(v_i)$ are the feasible candidate children of v_i. If yes, node v_i will be marked with *candidate forwarder* and be added into the *candidate forwarder set*. Also, we initially mark v_i with *uncovered* state and define $d^{\triangle}(v_i) = L$ for each non-sink node v_i.

(2) *Step 2: Candidate Forwarder Competition*

For any candidate forwarder v_i in the *candidate forwarder set*, we let $cost^*(v_i, CCS(v_i))$ denote the resulted optimal broadcasting cost when adopting Algorithm 2 on the topology $G_s(v_i, CCS(v_i))$, and define the following *competition factor*:

$$CF(v_i) = \frac{cost^*(v_i, CCS(v_i))}{|CCS(v_i)|}, \tag{11}$$

which characterizes the average resulted broadcasting cost on each covered node. The candidate forwarder v^* with the least *competition factor* in the *candidate forwarder set* will be selected as the competition winner. The winner v^* will be added into the *determined forwarder set* and then Algorithm 2 will be performed on the local topology $G_s(v^*, CCS(v^*))$ to get the forwarding decision of v^*.

(3) *Step 3: Information Update*

After the competition, any node $v_i \in CCS(v^*)$ will be marked with *covered* state, and each node $v_j \in CPS(v_i)$ will update its $CCS(v_j) = CCS(v_j) - \{v_i\}$. Suppose that $P_i^* = \{S_i^1, \ldots, S_i^M\}$ is the optimal solution when adopting Algorithm 2 on the topology $G_s(v^*, CCS(v^*))$, we will update $d^{\triangle}(v^*) = d(v^*, groupIR(S_i^1))$. Also, the *candidate forwarder set* will be updated. Specifically, any node $v_i \in CPS(v^*)$ will re-check whether it is the *candidate forwarder* and update the *candidate forwarder set* if necessary, since $CCS(v^*)$ turns to be $\emptyset$ that implies v^* will become the feasible candidate child of v_i. Further, we will identify all the *candidate forwarders* of which *candidate children set* are $\emptyset$, and then remove them from the *candidate forwarder set*. Afterwards, it skips to Step 2 to repeat the process until all the non-sink nodes are marked with *covered* states.

According to the above, we can find that the *Bottom-up Forwarder Selection Algorithm* is a iterative solution by rounds. Each round competition will determine a forwarder. Finally, all forwarders in the *determined forwarder set* and their forwarding decisions will constitute the broadcasting schedule. In this algorithm, we regard the *competition factor* in Eq. (11) as the metric to greedily select the forwarder at each round. Obviously, the candidate forwarder with the least *competition factor* will be preferred, this is because the greedy strategy that locally takes less cost to cover more sensing nodes each time could intuitively result in a lower total broadcasting cost of the network. Note that, we initially define $d^{\triangle}(v_i) = L$ for each non-sink node v_i, which implies there is no limitation on the deferred delay of v_i due to the fact that the deferred delay $D(v_i) - D^*(v_i)$ must be less than L for any broadcasting schedule under our model. Our algorithm follows the bottom-up design rule that any node v_i will not join the candidate forwarder competition until all nodes in $CCS(v_i)$ have been the feasible candidate children of v_i. In Step 3, we set $d^{\triangle}(v^*) = d(v^*, groupIR(S_i^1))$ after each round competition, to make sure v^*'s message reception time must precede the time that v^* first forwards the message.

Here, we will take the simple example in Fig. 6 to illustrate the above-mentioned *Bottom-up Forwarder Selection Algorithm*. In Fig. 6, we assume the tradeoff factor $\eta = 10$. For each sensor node v_i $(i \in \{1, \ldots, 10\})$, we will initially mark it with *uncovered* state

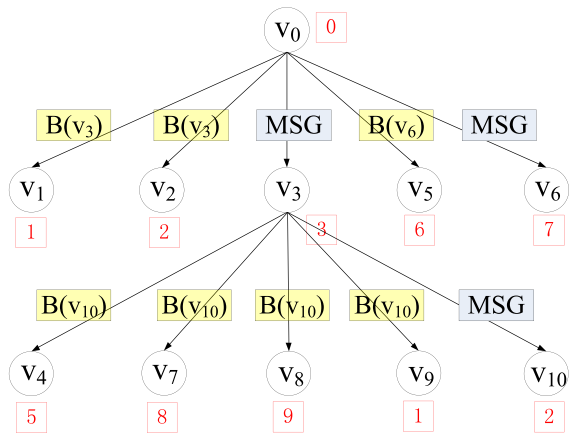


Fig. 7. Illustration of the solution to the example in Fig. 6.

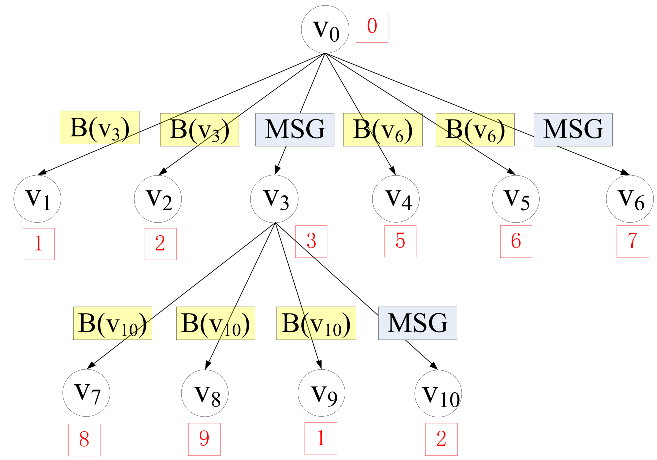


Fig. 8. Improved solution by schedule adjustment.

and set $d^{\triangle}(v_i)$ as the working-sleeping schedule period length 10. Obviously, node v_3 is the only *candidate forwarder* since all nodes in $CCS(v_3)$ are the feasible candidate children of v_3, which implies v_3 must be the competition winner and it will be added into the *determined forwarder set*. Algorithm 2 will then be performed on the topology $G_s(v_3, CCS(v_3))$ to get the solution $P_3^* = \{\{v_4, v_7, v_8, v_9, v_{10}\}\}$, which explicitly characterizes the forwarding decision of the determined forwarder v_3. Afterwards, we update $CCS(v_0) = \{v_1 . v_2 . v_3 . v_5, v_6\}$, $CCS(v_3) = \emptyset$, and $\{v_4, v_7, v_8, v_9, v_{10}\}$ will all be marked with *covered* states. Further, we update $d^{\triangle}(v_3) = d(v_3, v_{10}) = 9$, and the *candidate forwarder set* is updated with $\{v_0, v_1\}$ as v_3 has become the feasible candidate child of v_0 and v_1. It is obvious that $CF(v_0) = \frac{cost^*(v_0, CCS(v_0))}{|CCS(v_0)|} = \frac{24}{5} = 4.8$, which is less than $CF(v_1) = \frac{cost^*(v_1, CCS(v_1))}{|CCS(v_1)|} = \frac{11}{2} = 5.5$, thus v_0 will become the competition winner and be added into the *determined forwarder set*. Then, we will get v_0's forwarding decision $P_0^* = \{\{v_1, v_2, v_3\}, \{v_5, v_6\}\}$ by performing Algorithm 2 on the topology $G_s(v_0, CCS(v_0))$, and mark $\{v_1, v_2, v_3, v_5, v_6\}$ with *covered* states. Finally, we can get the broadcasting schedule that is represented by the *determined forwarders* $\{v_0, v_3\}$ and their forwarding decisions $P_0^* = \{\{v_1, v_2, v_3\}, \{v_5, v_6\}\}$, $P_3^* = \{\{v_4, v_7, v_8, v_9, v_{10}\}\}$. Fig. 7 exhibits the resulted broadcasting schedule.

4.2.3. Schedule adjustment

In *bottom-up forwarder selection* phase, we get the broadcasting schedule for our target network. However, we find that such broadcasting schedule could be further improved by changing the parent forwarders of some *DelayedReceivers*. Here, we still take Fig. 6 as example. In our solution as shown in Fig. 7, node v_4 is node v_3's *DelayedReceiver* with the deferred delay $D(v_4) - D^*(v_4) = d(v_4, v_{10}) = 7$. As shown in Fig. 8, if the *DelayedReceiver* v_4 chooses v_0 rather than v_3 as its parent forwarder, the deferred delay $D(v_4) - D^*(v_4)$ will be reduced to $d(v_4, v_6) = 2$ without affecting total energy consumption for broadcasting. This implies that our solution still offers the potential opportunity to improve the broadcasting delay performance by reassigning the parent forwarders to some *DelayedReceivers* from their *candidate parents sets*.

Based on the solution M^* that derived from *bottom-up forwarder selection* phase, we come up with the following schedule adjustment approach to further improve the broadcasting schedule:

Let $P_i^* = \{S_i^1, \ldots, S_i^M\}$ denote the forwarding decision of any determined forwarder v_i in M^*. For each *DelayedReceiver* v in M^*, we first check its any candidate parent $v_i \in CPS(v)$ that whether v_i is the determined forwarder in M^* and also $d(v_i, v) \leq d(v_i, groupIR(S_i^M))$, if yes, we will figure out $dd(v_i, v)$ by the following equation

$$dd(v_i, v) = \min_{S \in P_i^*} d(v, groupIR(S)); \tag{12}$$

If no, v_i will be removed from the candidate parents set $CPS(v)$ (Note that, we only consider any $v_i \in CPS(v)$ that satisfies $d(v_i, v) \leq d(v_i, groupIR(S_i^M))$ as the candidate parent of the *DelayedReceiver* v, this is because $d(v_i, v) > d(v_i, groupIR(S_i^M))$ can not make the *DelayedReceiver* v be covered by the forwarding decision P_i^*). Then, we will find the node v^* from $CPS(v)$, where

$$v^* = \arg \min_{v_i \in CPS(v)} dd(v_i, v). \tag{13}$$

If v^* is just the parent forwarder of node v in the solution M^*, no change will be performed on M^*; Otherwise, M^* will be adjusted by reassigning v^* as the parent forwarder of the *DelayedReceiver* v, it implies v will become the *DelayedReceiver* of the forwarder v^* with

$$IR(v) = groupIR\Big(\arg \min_{S \in P^*} d(v, groupIR(S)\Big), \tag{14}$$

where P^* denotes the forwarding decision of v^* in M^*.

Theorem 3. *For any low-duty-cycle sensor network $G = (V, E)$, our solution on G must have that*

$$\Delta_{delay} < \left(|V| - 1 - \lceil \frac{|V| - 1}{d_{max}} \rceil \right) \times L, \tag{15}$$

where $d_{\max}$ denotes the maximum node degree in G.

Proof. Our solution is to find a MDPT from the MDPFT to determine the broadcasting schedule, which implies it must have $D(v_i) - D^*(v_i) = d(v_i, IR(v_i)) < L$ for any *DelayedReceiver* v_i. In our solution, there are at least $\lceil \frac{|V|-1}{d_{\max}} \rceil$ *InstantReceivers*, which implies there are at most $|V| - 1 - \lceil \frac{|V|-1}{d_{\max}} \rceil$ *DelayedReceivers*. Let *DR* denote the set of the *DelayedReceivers* in our solution. Since $D(v_0) - D^*(v_0) = 0$ for the sink v_0 and $D(v_i) - D^*(v_i) = 0$ for any *InstantReceiver* v_i, we have $\Delta_{delay} = \sum_{v_i \in V} (D(v_i) - D^*(v_i)) = \sum_{v_i \in DR} (D(v_i) - D^*(v_i)) < |DR| \times L \leq (|V| - 1 - \lceil \frac{|V|-1}{d_{\max}} \rceil) \times L$. The proof is thus completed. □

Theorem 4. *Given any low-duty-cycle sensor network $G = (V, E)$, the time complexity of our solution on G is $O(|V|^2 \cdot d_{\max}^3)$, where $d_{\max}$ denotes the maximum node degree in G.*

Proof. (1) In fat-tree construction phase, we adopt an approach that is similar to the classical *Dijkstra Algorithm*, which will cost $O(|V|^2)$ time, to get the MDPFT. (2) In bottom-up forwarder selection phase, we can find that the running time is dominated by the computation time of the *competition factor* values for all nodes. At each round of candidate forwarder competition, each node v_i will first cost $O(d_{\max}^2)$ time to decide whether any node $v_j \in CCS(v_i)$ is the feasible candidate child of v_i, this implies that it will cost $O(d_{\max}^2) \times O(d_{\max}) = O(d_{\max}^3)$ time to decide whether v_i is the *candidate forwarder*. According to the analysis in Section 4.1, any *candidate forwarder* will take $O(d_{\max}^3)$ time to figure out its *competition factor* value. Thus, each node in V will totally cost at most $O(d_{\max}^3)$ running time to get the *competition factor* value at each round of competition. Due to the fact that it has at most $O(|V|)$ rounds of competition, each node in V will take at most $O(|V| \cdot d_{\max}^3)$ time in total. This implies the computation of the *competition factor* values for all nodes will totally cost $O(|V|) \times O(|V| \cdot d_{\max}^3) = O(|V|^2 \cdot d_{\max}^3)$ time. (3) In schedule adjustment phase, the running time is dominated by the computation time of v^* value (Eq. (13)) for all the *DelayedReceivers* in M^*. For each *DelayedReceiver* in M^*, it will cost $O(d_{\max})$ time to figure out v^*. As there are at most $O(|V|)$ *DelayedReceivers* in M^*, it must have totally $O(|V| \cdot d_{\max})$ time to figure out v^* for all the *DelayedReceivers* in M^*.

Overall, the time complexity of our solution on G will thus be $O(|V|^2) + O(|V|^2 \cdot d_{\max}^3) + O(|V| \cdot d_{\max}) = O(|V|^2 \cdot d_{\max}^3)$. The proof is thus completed. □

5. Discussion

As stated in Section 3.1, this paper assumes that the working-sleeping schedules of neighboring nodes are totally different from each other, which is usually true for most of the real low-duty-cycle WSNs. However, it could still have a small probability for the special case that very a few neighboring nodes could have an identical working-sleeping schedule. Suppose that the sender v_i has n+1 receivers $\{v_i^1, \ldots, v_i^{n+1}\}$, and receivers $\{v_i^1, \ldots, v_i^n\}$ have the same working-sleeping schedule, namely $d(v_i, v_i^j) = d(v_i, v_i^{j+1})$ $(1 \le j \le n-1)$ and $d(v_i, v_i^n) < d(v_i, v_i^{n+1})$. If we adopt the traditional solution, the total energy consumption for broadcasting $E_a = 2 * k * e_s^d + (n+1) * k * e_r^d$; If the sender v_i defers the message reception time of all the *DelayedReceivers* $\{v_i^1, \ldots, v_i^n\}$ to the *active slot* of the *InstantReceiver* v_i^{n+1}, the total energy consumption for broadcasting will be $E_b = e_s^b + n * e_r^b + k * e_s^d + (n+1) * k * e_r^d$. Thus, we can find that the energy benefit from deferring will be $E_a - E_b = k * e_s^d - (e_s^b + n * e_r^b)$. Obviously, if very a few neighboring nodes have an identical working-sleeping schedule, *i.e.,* n is less, the energy benefit $E_a - E_b$ will still be greater than 0. For the application with large broadcasting message, such energy benefit will be more significant. This implies the conclusion that *the total energy consumption for broadcasting can be essentially characterized by the number of the* InstantReceivers will be still true even for this special case. Accordingly, our proposed solution is still applicable for the special case where very a few neighboring nodes could have an identical working-sleeping schedule.

Here, we can extend our solution mentioned above to this special case by regarding the neighboring nodes with the same working-sleeping schedule as one *virtual node*. For example, given a single-hop topology $G_s(\{v_i\} \bigcup \{v_{j_1}, \ldots, v_{j_6}\}, E_s)$ in which $d(v_i, v_{j_1}) < d(v_i, v_{j_2}) = d(v_i, v_{j_3}) = d(v_i, v_{j_4}) < d(v_i, v_{j_5}) = d(v_i, v_{j_6})$, we can re-mark receivers $\{v_{j_1}, \ldots, v_{j_6}\}$ as $\{v_i^1, v_i^2, v_i^3\}$, where $v_i^1 = \{v_{j_1}\}$, $v_i^2 = \{v_{j_2}, v_{j_3}, v_{j_4}\}$, $v_i^3 = \{v_{j_5}, v_{j_6}\}$. Here, each v_i^j $(j \in \{1, 2, 3\})$ denotes a *virtual node*, we can define that $t(v_i^1) = t(v_{j_1})$, $t(v_i^2) = t(v_{j_2}) = t(v_{j_3}) = t(v_{j_4})$ and $t(v_i^3) = t(v_{j_5}) = t(v_{j_6})$. Further, a *virtual node* is called the *DelayedReceiver*/*InstantReceiver* if and only if all sensor nodes in this *virtual node* are the *DelayedReceivers*/*InstantReceivers*. Note that, any *InstantReceiver virtual node* represents one broadcasting message transmission. For any *coverage group* $\{v_i^j, \ldots, v_i^k\}$ $(j \le k)$, $d_{sum}(\{v_i^j, \ldots, v_i^k\})$ will be represented by

$$d_{sum}\left(\{v_i^j, \ldots, v_i^k\}\right) = \sum_{m=j}^{k} \sum_{v \in v_i^m} d\left(v, v_i^k\right) = \sum_{m=j}^{k} \left(|v_i^m| \times d\left(v_i^m, v_i^k\right)\right) \tag{16}$$

where $|v_i^m|$ denotes the number of sensor nodes in *virtual node* v_i^m.

Except the transmission of broadcasting messages, the commonly seen traffics in WSNs are mainly from periodical data collection applications and rare-event detection applications. For the periodical data collection applications, we can definitely be aware of the period of data collection, which is usually a fixed relatively long duration. Accordingly, we can carefully pick a proper time to start broadcasting from the sink, so that the collisions between the broadcasting messages and data collection messages can be avoided. As the rare-event detection applications usually have low traffics in practice, the collision probability between the broadcasting messages and event detection messages will be relatively low, we can thus simply adopt the commonly-used traditional CSMA/CA protocol to well solve such collisions.

In this paper, we do not consider the ACK packets. Actually, we can find the conclusion that *the total energy consumption for broadcasting can be essentially characterized by the number of InstantReceivers* will still hold even the energy consumption of ACK packets are considered into the energy consumption model. When considering the ACK packets, note that, we require *DelayedReceivers* should reply the ACK packets to the sender after they *overhear* the message.

In Section 3.1, we simply assume that every time slot is set long enough so that it can accommodate the transmission of any potential broadcasting message. Such assumption is applicable in practice, however, could bring inefficient delay performance for other traffics. Specifically, the broadcasting applications with large message, such as code update, require that each time slot should be set long enough. which will incur poor delay performance for those commonly-seen applications with one packet transmission, *e.g.,* configuration distribution, data collection and rare-event detection. In practice, we can thus set the length of each time slot so that it can accommodate the transmission of at least one packet, to guarantee the delay performance for the applications with one packet transmission. For some applications with large message, we can transmit each packet in the message one by one in a timely pipeline way.

In our solution, each node will initially keep awake immediately after the deployment and the sink will derive the network topology according to some existing solution. Based on the network topology, the sink will execute our algorithm to obtain the broadcasting schedule and then distribute it to all nodes in the network, and this will be done during the initialization phase of the network and is an one-time task. Actually, this is also the commonly used implementation way for most of the existing centralized algorithms. Once getting the broadcasting strategy, each node will put itself into the low-duty-cycle mode according to its own working-sleeping schedule, which is based on a particular power management protocol. In practice, the topology of a network may change in three cases: (1) an existing link disappears due to obstacles; (2) a new link appears due to removal of obstacles; (3) a node dies. In all cases, a simple scheme is to periodically re-execute our algorithm to update the broadcasting schedule according to the history record about the frequency of topology change, this is also the commonly used approach for almost all of the existing works. Actually, how to make a low-cost and high-efficient on-line adaptive scheme to overcome the topology change is still an open problem for all of the existing works, and we plan to further study this problem in our future work.

In low-duty-cycle WSNs, we usually improve the network performance, *e.g.*, to minimize average detection delay, by carefully designing the working schedules of all nodes to make the neighboring nodes rotate the sensory coverage [41], which implies the neighboring nodes almost have the different working-sleeping schedules from each other. Moreover, for a specific network performance requirement, it is usual that each node will correspondingly increase the period length L as the node density increases.

6. Performance evaluation

6.1. Simulation settings

Here, we will evaluate the performance of proposed solution by extensive simulations. Suppose that there are *Num* sensor nodes are uniformly deployed in a square sensory field with the size of 100 m*100 m, where the sink node is located at the center. Also, we assume that each working-sleeping schedule period of any node consists of one *active slot* and $L-1$ *sleeping slots*, and each node independently and randomly determines its periodical working-sleeping schedule. In our setting, all nodes are assumed have an identical communication range r_c. Unless otherwise stated, we set Num=800, L=200, r_c=15 m, η=200, and all the experiment results are generated by averaging over 20 times.

In our simulations, note that, we allow the neighboring nodes could have an identical working-sleeping schedule. However, the parameters set in our simulations can make very a few neighboring nodes, or even no neighboring nodes, have the same working-sleeping schedules, which can well-simulate the real low-duty-cycle sensor networks.

6.2. Baselines

In this section, we will take the following 7 heuristic approaches as the baselines to evaluate the performance of our proposed MDPFT-based bottom-up solution.

- **MDPT-based delay-first solution:** This approach adopts a delay-first strategy without deferring. Specifically, it first finds a MDPT, which is rooted at the sink node, from the network topology, then the sink node broadcasts the message directly along with the MDPT. In this solution, all sensing nodes are the *InstantReceivers* and any local single-hop broadcast will be implemented by multiple unicasts.
- **MDPT-based energy-first solution:** This approach adopts an energy-first strategy where each forwarder only sets exactly one of its receivers as the *InstantReceiver*. It first finds a MDPT rooted at the sink node from the network topology, then the sink node will forward the message along with the MDPT in a top-down order. Let $\{v_i^1, \ldots, v_i^N\}$ denote the children of any forwarder v_i on MDPT where $d(v_i, v_i^j) < d(v_i, v_i^{j+1})$ $(1 \leq j \leq N-1)$, any forwarder v_i will check whether its deferred delay $D(v_i) - D^*(v_i) \geq d(v_i, v_i^N)$ once its parent has determined the forwarding decision, if yes, v_i will make the forwarding decision that nodes $\{v_i^1, \ldots, v_i^N\}$ all defer their message reception time to the *active slot* of v_i^1's next working-sleeping schedule period; otherwise, v_i will make the forwarding decision $P_i^* = \{\{v_i^1, \ldots, v_i^N\}\}$. In other words, there is only one time message transmission for any forwarder on MDPT.
- **MDPFT-based delay-first solution** [14]: This approach is similar to the MDPT-based delay-first solution, the difference is that this approach determines the broadcasting schedule from the MDPFT by adopting a top-down and iterative greedy strategy. The detailed description of this approach is shown in [14].
- **MDPFT-based energy-first solution** [14]: This approach is similar to the MDPT-based energy-first solution, the difference is that this approach determines the broadcasting schedule from the MDPFT by adopting a top-down and iterative greedy strategy. The detailed description of this approach is shown in [14].
- **MDPT-based top-down solution** [14]: This approach first finds a MDPT rooted at the sink node from the network topology, and then determines the forwarding decisions of all forwarders on MDPT in a top-down order. Specifically, any forwarder on MDPT will make the forwarding decision only if its parent has determined the forwarding decision. Once $D(v_i) - D^*(v_i)$ is determined, any forwarder v_i on MDPT will make the forwarding decision based on the constraint that $D(v_i) - D^*(v_i)$ must be less than the sleeping latency from v_i to its first *InstantReceiver*. The detailed description of this approach is shown in [14].
- **MDPFT-based top-down solution** [14]: This approach is similar to the MDPT-based top-down solution, the difference is that this approach determines the broadcasting schedule from the MDPFT by adopting a top-down and iterative greedy strategy. The detailed description of this approach is shown in [14].
- **MDPT-based bottom-up solution:** This approach first finds a MDPT rooted at the sink node from the network topology, and then determines the forwarding decisions of all forwarders on MDPT in a bottom-up order. Specifically, any forwarder on MDPT will make the forwarding decision only if any of its children is either the leaf node or the forwarder that has determined the forwarding decision. Initially, this approach defines $d^{\triangle}(v) = L$ for each sensing node v, and let $\{v_i^1, \ldots, v_i^N\}$ denote the children of any forwarder v_i on MDPT where $d(v_i, v_i^j) < d(v_i, v_i^{j+1})$ $(1 \leq j \leq N-1)$, any forwarder v_i will perform Algorithm 2 on the topology $G_s(\{v_i\} \bigcup \{v_i^1, \ldots, v_i^N\}, E_s)$ to get the forwarding decision $P_i^* = \{S_i^1, \ldots, S_i^M\}$ once any of its children is either the leaf node or the forwarder that has determined the forwarding decision, and then update its $d^{\triangle}(v_i) = d(v_i, groupIR(S_i^1))$.

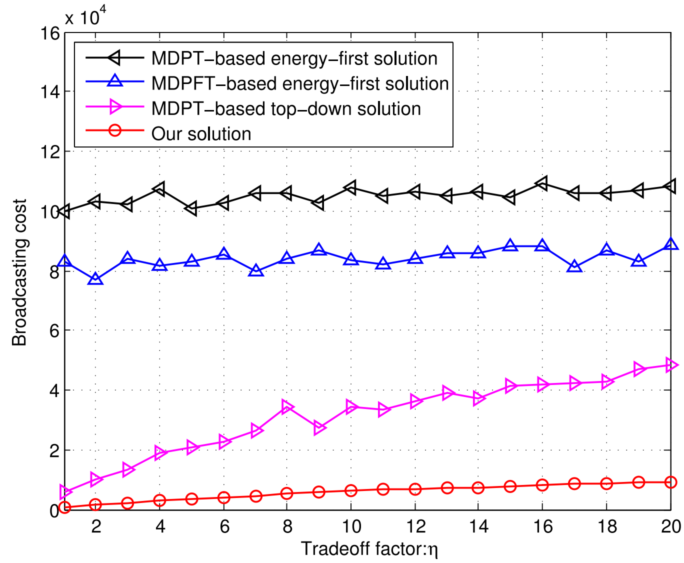


Fig. 9. Performance comparison when η varies between 1 and 20.

6.3. The impact of tradeoff factor

Next, we will compare our proposed MDPFT-based bottom-up solution with the above baselines under various configurations. First, we will simulate the applications with various broadcasting performance requirements by adjusting the tradeoff factor η. Figs. 9 and10 exhibit the performance comparison on our solution and the baselines when η varies from 1 to 20. Under this con-

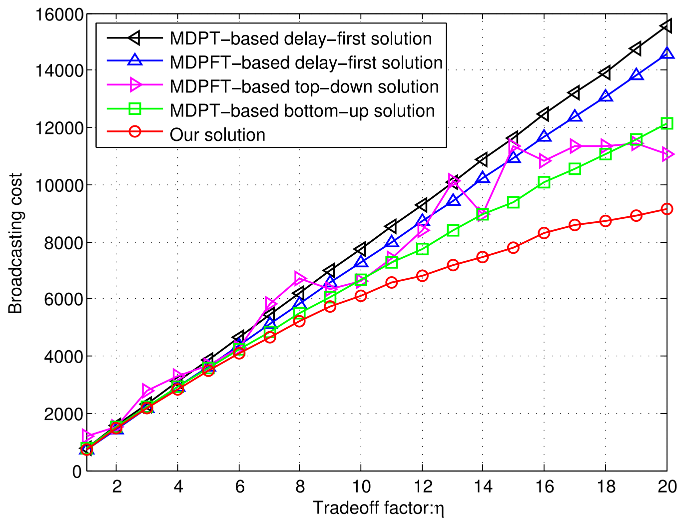


Fig. 10. Performance comparison when η varies between 1 and 20.

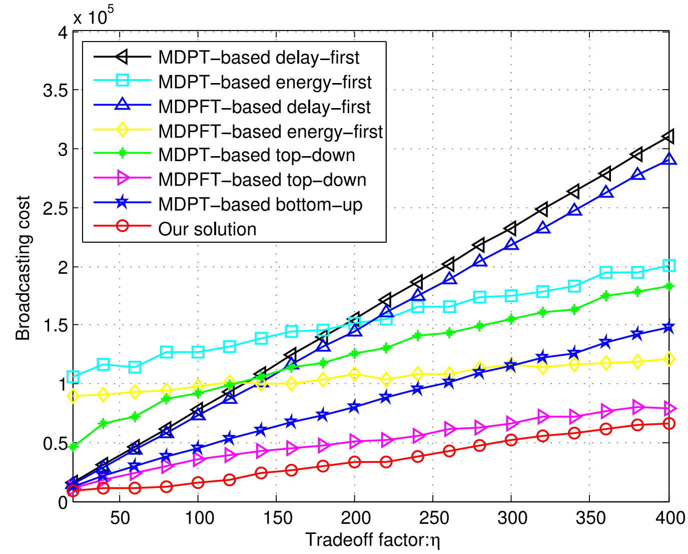


Fig. 11. Performance comparison when η varies between 20 and 400.

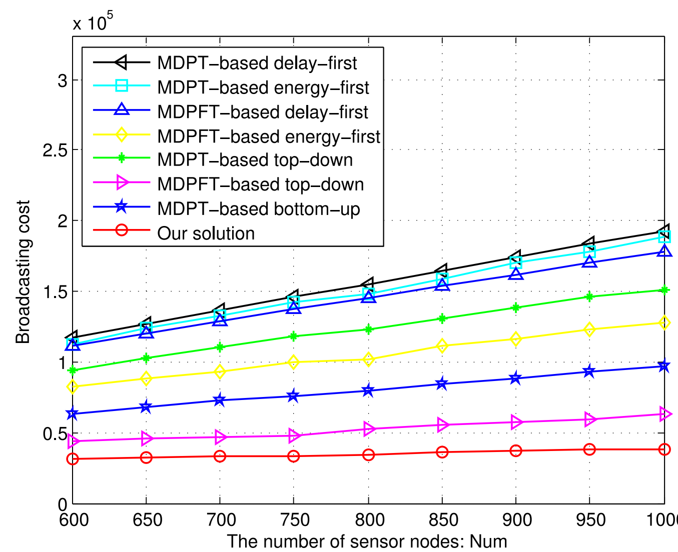


Fig. 12. Performance comparison when *Num* varies between 600 and 1000.

figuration, our solution has a much better performance than the energy-first strategies and has a similar performance to the delay-first strategies when η is small, since small η implies the broadcasting cost will be dominated by the delay performance. As η increases, we can find that the performance advantage of our solution is getting larger, since the increase on η will make the energy performance have a larger impact on the broadcasting cost, which implies the delay-first strategies that only focus on the delay performance rather than the tradeoff between delay performance and energy performance will gradually lose their advantages. Also, our solution performs better than all the top-down solutions, this is because the top-down solutions could bring large broadcasting delay. For the top-down solutions, specifically, any forwarder v_i will defer the message reception time of all its receivers $\{v_i^1, \ldots, v_i^N\}$ to the *active slot* of v_i^1's next working-sleeping schedule period if $D(v_i) - D^*(v_i) \geq d(v_i, v_i^N)$, which will make the deferred delay of v_i^1 be increased to L and the other receivers also significantly increase their deferred delay, and the constraint that $D(v_i) - D^*(v_i)$ must be less than the sleeping latency from v_i to its first *InstantReceiver* will also degrade the delay performance. However, our bottom-up solution can make sure that any *DelayedReceiver* has low deferred delay which must be less than L, and the schedule adjustment scheme will also improve the delay performance. Our solution also outperforms the MDPT-based bottom-up solution, this is because our solution is based on the MDPFT which will provide a higher flexibility on the forwarders selection.

Fig. 11 shows the performance comparison between various solutions when η varies from 20 to 400. We can find that our solution still has the best performance over all the solutions even if η is large. As η is getting larger, the energy performance will gradually dominate the broadcasting cost, our solution will thus have a larger performance advantage over the MDPFT-based delay-first solution but a smaller performance advantage over the MDPFT-based energy-first solution. According to the simulation result, we find the performance of our solution will be close to that of the MDPFT-based top-down solution as η is large enough, this is because our proposed bottom-up solution will result in the limitation on energy performance. For any forwarder v_i in our solution, specifically, the bound $d^{\Delta}(v_i)$ implies the constraint that there must be at least one *InstantReceiver* in the constrained range $CR(v_i)$, which will limit the improvement of energy performance. Thus, our solution could not have a significant advantage compared with the MDPFT-based top-down solution when η is large enough. However, our solution still exhibits a much better performance than the MDPT-based bottom-up solution even if η is large, and its performance advantage over the MDPT-based bottom-up solution will get larger as η increases. This is because our solution has a better flexibility on the forwarders selection to improve the energy performance, compared with the MDPT-based bottom-up solution.

6.4. The impact of network density

Further, we evaluate the impact of network density on broadcasting cost. Fig. 12 shows the performance comparison between various solutions when $\eta = 200$ and *Num* varies from 600 to 1000. We can find that no matter how the network density varies, our solution will always outperform the other solutions, and also the performance advantage of our solution over the MDPFT-based top-down solution will get larger as the network density increases.

6.5. The impact of duty cycle

Fig. 13 exhibits the impact of duty cycle on broadcasting cost. We can find that as L increases, *i.e.*, duty cycle decreases, the per-

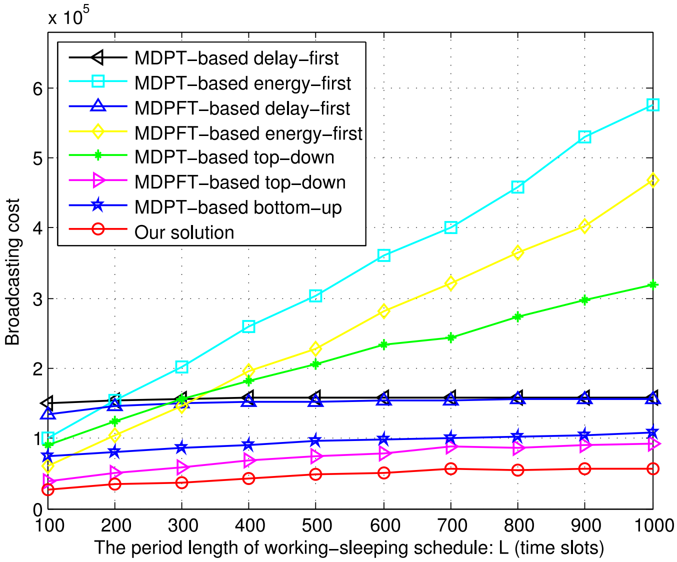


Fig. 13. Performance comparison when L varies between 100 and 1000.

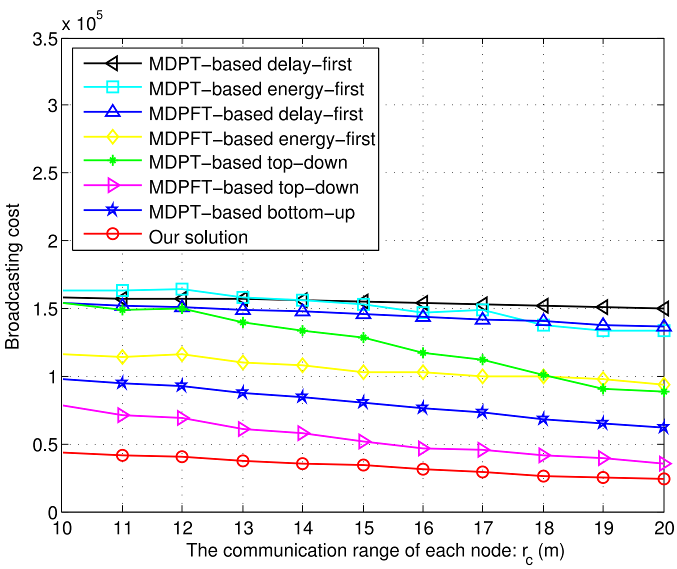


Fig. 14. Performance comparison when r_c varies between 10 and 20.

formance of the delay-first strategies will gradually turn to be stable, since the broadcasting cost resulted from the delay-first strategies mainly depends on the number of message transmissions, which will converge to the number of all sensing nodes as L increases. Also, we find the performance of the energy-first strategies will significantly increase as L increases. In Fig. 13, we find that our solution will always have the best performance over all the solutions under whatever duty cycle, and our solution will have a larger performance advantage over the top-down strategies as L increases, this is because when η is fixed, the increase of L will make delay performance have a larger impact on the broadcasting cost, and as stated before, our solution will exhibit the better delay performance compared with the top-down strategies.

6.6. The impact of communication range

In Fig. 14, we show the relationship between broadcasting cost and the communication range r_c. In general, the broadcasting cost of all the solutions will decrease as r_c increases, this is because the increase on r_c will basically reduce the number of forwarders and also make any forwarder have more receivers with the identical working-sleeping schedule. Here, our solution still exhibits the best performance over all the solutions no matter how r_c varies. Note that, our solution will have a larger performance advantage over the top-down strategies as r_c decreases.

Therefore, we can conclude that our solution always outperforms the other solutions under whatever configurations.

7. Conclusion and future work

In this paper, we focus on the broadcast problem for low-duty-cycle WSNs, and adopt a novel opportunistic broadcasting transmission model, which provides a flexible control on the tradeoff between delay performance and energy performance for broadcasting. We define a generalized broadcasting cost function, which can provide an adaptive control on the tradeoff between average broadcasting delay and total energy consumption for broadcasting to meet various performance requirements. Our goal is to utilize the opportunistic broadcasting transmission model to find an efficient broadcasting schedule to minimize such broadcasting cost function, so that the specific broadcasting performance requirement is satisfied. To this end, we first define the receiver-constrained minimum cost single-hop broadcast problem, which can be solved by a dynamic programming algorithm with a polynomial running time. Then, we extend it to our target problem, and come up with a novel MDPFT-based bottom-up solution. The simulation results reveal that our proposed bottom-up solution significantly outperforms the existing top-down solutions and the other solutions.

In our future work, we aim to consider how to extend our target problem to unreliable networks. In practice, many WSNs are usually deployed in a tough environment with lossy links. The combination of low-duty-cycle operation and the unreliability of links will further exacerbate the inefficiency of broadcasting, which implies our target problem under the unreliable networks has become a challenging issue. To this end, how to carefully joint the opportunistic broadcasting transmission model and link correlation to design efficient broadcasting algorithms will be our main concern in future. Besides, our future work will further consider how to combine with the Wake-On-Radio model, which can employ a second low power radio as a trigger to wake up the primary radio, and how to extend our target problem to asynchronous low-duty-cycle networks without any time synchronization.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61502251, 61572263, 61472193, 61502249, 61502229, 61672276), China Postdoctoral Science Foundation Funded Project (No. 2016M601859), the Post-Doctoral Fund of Jiangsu Province (No. 1701047A), Natural Science Foundation of Jiangsu Province (No. BK20150846).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.comnet.2018.03.007.

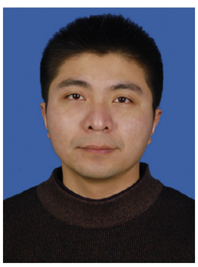
References

[1] Y. Liu, A. Liu, S. Guo, Z. Li, Y.J. Choi, H. Sekiya, Context-aware collect data with energy efficient in cyber-physical cloud systems, Future Gener. Comput. Syst. (2017), doi:10.1016/j.future.2017.05.029.
[2] X. Liu, S. Zhao, A. Liu, N. Xiong, A.V. Vasilakos, Knowledge-aware proactive nodes selection approach for energy management in internet of things, Future Gener. Comput. Syst. (2017), doi:10.1016/j.future.2017.07.022.
[3] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, X. Li, Does wireless sensor network scale? a measurement study on greenorbs, IEEE Trans. Parallel Distrib. Syst. (TPDS) 24 (10) (2013) 1983–1993.

[4] Y. Liu, X. Mao, Y. He, K. Liu, W. Gong, J. Wang, CitySee: not only a wireless sensor network, IEEE Netw. (2013) 42–47.
[5] F. Li, Y. Yang, Z. Chi, L. Zhao, Y. Yang, J. Luo, Trinity: enabling self-sustaining WSNs indoors with energy-free sensing and networking, ACM Transactions on Embedded Computing Systems 17 (2) (2018) 57:1–57:27.
[6] M. Li, Z. Yang, Y. Liu, Sea depth measurement with restricted floating sensors, ACM Trans. Embed. Comput. Syst. (TECS) 13 (1) (2013) 1–21. Article No. 1.
[7] P. Guo, T. Jiang, Q. Zhang, K. Zhang, Sleep scheduling for critical event monitoring in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. (TPDS) 23 (2) (2012) 345–352.
[8] G. Anastasi, M. Conti, M.D. Francesco, Extending the lifetime of wireless sensor networks through adaptive sleep, IEEE Trans. Ind. Inf. 5 (3) (2009) 351–365.
[9] F. Liu, C.Y. Tsui, Y.J. Zhang, Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks, IEEE Trans. Wirel. Commun. (TWC) 9 (7) (2010) 2258–2267.
[10] Z. Li, M. Li, Y. Liu, Towards energy-fairness in asynchronous duty-cycling sensor networks, in: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), 2012, pp. 801–809.
[11] M. Stemm, R.H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, IEICE Trans. Commun. E80-B (8) (1997) 1125–1131.
[12] Z. Li, M. Li, J. Liu, S. Tang, Understanding the flooding in low-duty-cycle wireless sensor networks, in: Proceedings of the IEEE International Conference on Parallel Processing (ICPP), 2011, pp. 673–682.
[13] Y. Gu, T. He, Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links, in: Proceedings of the Fifth ACM International Conference on Embedded Networked Sensor Systems (SenSys), 2007, pp. 321–334.
[14] L. Xu, G. Yang, L. Wang, J. Xu, B. Wang, Opportunistic broadcasting for low-power sensor networks with adaptive performance requirements, Wirel. Netw. (2017) 1–21, doi:10.1007/s11276-017-1473-6.
[15] S. Guo, Y. Gu, B. Jiang, T. He, Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links, in: Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom), 2009, pp. 133–144.
[16] J. Lu, K. Whitehouse, Flash flooding: exploiting the capture effect for rapid flooding in wireless sensor networks, in: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), 2009, pp. 2491–2499.
[17] D. Zhao, K.W. Chin, R. Raad, Approximation algorithms for broadcasting in duty cycled wireless sensor networks, Wirel. Netw. 20 (8) (2014) 2219–2236.
[18] X. Jiao, W. Lou, X. Wang, J. Ma, J. Cao, X. Zhou, On interference-aware gossiping in uncoordinated duty-cycled multi-hop wireless networks, Ad Hoc Netw. 11 (4) (2013) 1319–1330.
[19] W. Du, J.C. Liando, H. Zhang, M. Li, When pipelines meet fountain: fast data dissemination in wireless sensor networks, in: Proceedings of the Thirteenth ACM International Conference on Embedded Networked Sensor Systems (SenSys), 2015, pp. 365–378.
[20] R. Zhang, H. Moungla, J. Yu, L. Chen, A. Mehaoua, Multi-channel broadcast in asymmetric duty cycling wireless body area networks, in: Proceedings of the IEEE International Conference on Communications (ICC), 2017, pp. 1–6.
[21] D.T. Le, T.L. Duc, V.V. Zalyubovskiy, D.S. Kim, H. Choo, Collision-tolerant broadcast scheduling in duty-cycled wireless sensor networks, J. Parallel Distrib. Comput. (JPDC) 100 (C) (2017) 42–56.
[22] T.L. Duc, D.T. Le, V.V. Zalyubovskiy, D.S. Kim, H. Choo, Towards broadcast redundancy minimization in duty-cycled wireless sensor networks, Int. J. Commun. Syst. 30 (6) (2017) 1–21.
[23] C. Jiang, T.S. Li, J.B. Liang, H. Wu, Low-latency and energy-efficient data preservation mechanism in low-duty-cycle sensor networks, Sensors 17 (5) (2017) 1–17.
[24] J. Niu, L. Cheng, Y. Gu, J. Jun, Q. Zhang, Minimum-delay and energy-efficient flooding tree in asynchronous low-duty-cycle wireless sensor networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2013, pp. 1261–1266.
[25] L. Cheng, Y. Gu, T. He, J. Niu, Dynamic switching-based reliable flooding in low-duty-cycle wireless sensor networks, in: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), 2013, pp. 1393–1401.
[26] X. Zhang, X. Jia, J. Jin, K.S. Dan, Delay-constrained efficient broadcasting in duty-unaware asynchronous wireless sensor networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2015, pp. 1841–1846.
[27] S. Guo, S.M. Kim, T. Zhu, Y. Gu, T. He, Correlated flooding in low-duty-cycle wireless sensor networks, in: Proceedings of the Nineteenth IEEE International Conference on Network Protocols (ICNP), 2011, pp. 383–392.
[28] I. Jemili, D. Ghrab, A. Belghith, M. Mosbah, Context-aware broadcast in duty–cycled wireless sensor networks, Int. J. Semant. Web Inf. Syst. (IJSWIS) 13 (3) (2017) 48–67.
[29] K. Han, J. Luo, L. Xiang, M. Xiao, L. Huang, Achieving energy efficiency and reliability for data dissemination in duty-cycled WSNs, IEEE/ACM Trans. Netw. (TON) 23 (4) (2015) 1041–1052.
[30] Z. Zhao, W. Dong, J. Bu, Y. Gu, C. Chen, Link-correlation-aware data dissemination in wireless sensor networks, IEEE Trans. Ind. Electron. 62 (9) (2015) 5747–5757.
[31] Q. Chen, S. Cheng, H. Gao, J. Li, Z. Cai, Energy-efficient algorithm for multicasting in duty-cycled sensor networks, Sensors 15 (12) (2015) 31224–31243.
[32] T.L. Duc, D.T. Le, V.V. Zalyubovskiy, D.S. Kim, H. Choo, Level-based approach for minimum-transmission broadcast in duty-cycled wireless sensor networks, Pervasive Mob. Comput. 27 (C) (2016) 116–132.
[33] L. Xu, X. Zhu, H. Dai, X. Wu, G. Chen, Towards energy-fairness for broadcast scheduling with minimum delay in low-duty-cycle sensor networks, Comput. Commun. (COMCOM) 75 (2016) 81–96.
[34] L. Xu, G. Chen, J. Cao, S. Lin, H. Dai, X. Wu, F. Wu, Optimizing energy efficiency for minimum latency broadcast in low-duty-cycle sensor networks, ACM Trans. Sensor Netw. (TOSN) 11 (4) (2015) 1–31. Article No. 57
[35] T.L. Duc, D.T. Le, V.V. Zalyubovskiy, D.S. Kim, H. Choo, Towards broadcast redundancy minimization in duty-cycled wireless sensor networks, Int. J. Commun. Syst. 32 (2) (2016) 131–138.
[36] L. Cheng, J. Niu, Y. Gu, C. Luo, T. He, Achieving efficient reliable flooding in low-duty-cycle wireless sensor networks, IEEE/ACM Trans. Netw. (TON) 24 (6) (2016) 3676–3689.
[37] Z. Cao, J. Wang, D. Liu, X. Zheng, Chase: taming concurrent broadcast for flooding in asynchronous duty cycle networks, in: Proceedings of the IEEE Twenty Fourth International Conference on Network Protocols (ICNP), 2016, pp. 1–10.
[38] S. Wu, J. Niu, W. Chou, M. Guizani, Delay-aware energy optimization for flooding in duty-cycled wireless sensor networks, IEEE Trans. Wirel. Commun. (TWC) 15 (12) (2016) 8449–8462.
[39] F. Wang, J. Liu, On reliable broadcast in low duty-cycle wireless sensor networks, IEEE Trans. Mob. Comput. (TMC) 11 (5) (2012) 767–779.
[40] M. Marti, B. Kusy, G. Simon, A. Ldeczi, The flooding time synchronization protocol, in: Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 39–49.
[41] Q. Cao, T. Abdelzaher, T. He, J. Stankovic, Towards optimal sleep scheduling in sensor networks for rare-event detection, in: Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN), 2005, pp. 20–27.
[42] Q. Wang, M. Hempstead, W. Yang, A realistic power consumption model for wireless sensor network devices, in: Proceedings of the Third Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON), 2006, pp. 286–295.

Lijie Xu is currently an assistant professor of the School of Computer Science at Nanjing University of Posts and Telecommunications, Jiangsu, China. He received the Ph.D. degree in the Department of Computer Science and Technology from Nanjing University, Jiangsu, China, in 2014. He was a research assistant in the Department of Computing at the Hong Kong Polytechnic University from Nov. 2011 to May 2012. His research interests are mainly in the areas of wireless sensor networks, ad-hoc networks, mobile and distributed computing.

Geng Yang received the Ph.D. degree in Computational Mathematics from Laval University in 1994 and was a post-doctoral research fellow in Center for Research on Computation & its Applications at University of Montreal from 1994 to 1996 in Canada respectively. His research interest includes information & network security, privacy preserving, and parallel & distributed computing. Professor Yang is a member of the IEEE Computer Society.



Jia Xu received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the Ph.D. degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in Nanjing University of Posts and Telecommunications. He was a visiting scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov. 2014 to May 2015. His main research interests include crowdsourcing, opportunistic networks and wireless sensor networks.

Lei Wang received his bachelor degree from Anhui University of China in 2008. He was a Master-Doctor combined program graduate student in computer science of Nanjing University of Science & Technology (NUST), China since 2008. He was a visiting student in Michigan State University during 2013–C2014. Currently, he is an assistant professor in School of Computer, Nanjing University of Posts and Telecommunications, China. His research interests are network coding, wireless sensor network, multipath routing, distributed storage system.

Haipeng Dai received the B.S. degree in the Department of Electronic Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the Ph.D. degree in the Department of Computer Science and Technology in Nanjing University, Nanjing, China, in 2014. His research interests are mainly in the areas of wireless charging, mobile computing and data mining. He is a research assistant professor in the Department of Computer Science and Technology in Nanjing University. His research papers have been published in many prestigious conferences and journals such as ACM MobiHoc, ACM VLDB, ACM SIGMETRICS, IEEE INFOCOM, IEEE ICDCS, IEEE ICNP, ICPP, IEEE IPSN, IEEE JSAC, IEEE TON, IEEE TMC, IEEE TPDS, IEEE TSC and IEEE TOSN. He is an IEEE and ACM member. He serves/ed as Poster Chair of the IEEE ICNP'14, TPC member of the IEEE ICNP'14, IEEE ICC'14-17, IEEE ICCCN'15-17 and the IEEE Globecom'14-17. He received Best Paper Award from IEEE ICNP'15, Best Paper Award Candidate from IEEE INFOCOM'17.