

Energy-efficient Broadcast Scheduling with Minimum Latency for Low-Duty-Cycle Wireless Sensor Networks

Lijie Xu*, Jiannong Cao[†], Shan Lin[‡], Haipeng Dai*, Xiaobing Wu* and Guihai Chen*[§]

*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China

[†]Internet and Mobile Computing Lab, The Hong Kong Polytechnic University, Hong Kong

[‡]Department of Computer and Information Sciences, Temple University, Philadelphia, USA

[§]Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, Shanghai 200240, China
Email: {ljxu83, dhpphd2003}@gmail.com, csjcao@comp.polyu.edu.hk, shan.lin@temple.edu, {wuxb, gchen}@nju.edu.cn

Abstract—For low-duty-cycle wireless sensor networks, multihop broadcasting is a challenging problem, since every node has its own working schedules. In this paper, we design a novel broadcasting algorithm, of which key idea is to let some early wake-up nodes postpone their wake-up slots to overhear broadcasting message from its neighbors. This design utilizes the spatiotemporal locality of broadcasting to reduce the number of transmissions. We show that to find the broadcasting schedule with minimal latency and optimized total energy consumption is NP-hard, and then design an approximation algorithm that can guarantee the optimality of broadcasting latency and achieve a polylogarithmic approximation ratio for total energy consumption. Compared with the traditional solution, extensive experimental results show that our algorithm achieves the minimal broadcasting latency while reducing energy consumption significantly.

Keywords—low-duty-cycle WSNs; broadcast scheduling; energy efficient; minimal latency

I. INTRODUCTION

It is important and challenging to minimize the latency and the energy consumption of the broadcasting in Wireless Sensor Networks (WSNs), especially for low-duty-cycle WSNs, in which every sensor node has its own working schedule to wake-up periodically to perform sensing and communication tasks. Compared with the always-awake networks, low-duty-cycle WSNs usually yield a notable increase on communication latency due to the periodic sleeping, and thus latency is always taken as the first consideration in such networks.

In this work, we focus on the following problem: how to design a broadcasting schedule that can achieve the minimal latency while reducing total energy consumption for low-duty-cycle WSNs. Existing solutions [1]–[6] for broadcasting in low-duty-cycle WSNs usually implement broadcast with multiple unicasts, which is energy-inefficient especially for applications of large message broadcasting, such as *code updates*. Actually, the broadcast nature of wireless communication offers opportunities to reduce the total number of transmissions in broadcasting, even for duty-cycled networks where every node has its own schedule. To improve the energy efficiency of broadcasting, some nodes

may postpone their wake-up slots to receive broadcasting message, increasing their latency. But these nodes can be carefully selected so that they are not on latency-critical paths. Therefore, their schedule changes do not affect the minimal broadcasting latency.

The contributions of this work are as follows:

- By capturing the spatiotemporal characteristic of multihop broadcasting, we model our scheduling problem as the Latency-optimal Group Steiner Tree Problem, which is proved to be NP-hard.
- We propose a novel wake-up schedule to realize broadcasting for low-duty-cycle WSNs by utilizing spatiotemporal locality of the broadcasting, and also show that this design can guarantee the optimality of broadcasting latency and achieve an approximation ratio of $O(\log N \cdot \log d_{max})$ in terms of total energy consumption, where N and d_{max} denote the number of sensor nodes and the maximum node degree respectively.
- Extensive simulations results show that our solution makes a significant improvement compared with the traditional solution.

The rest of the paper is organized as follows: Section II illustrates the network model and states the problem. Detailed description of our proposed scheme is presented in Section III. Followed by the simulation results in Section IV, we conclude the paper in Section V.

II. NETWORK MODEL AND PROBLEM FORMULATION

A. Network Model and Assumptions

In this paper, we assume that N sensor nodes are uniformly deployed in a circular sensory field with a radius of R and the sink node is located at the center of the sensory field. Also, it is assumed that time is divided into a number of equal time slots and each time slot is set long enough so that it can accommodate the transmission of the potential large broadcasting message. Each time slot is either in *sleep state* or in *active state*. In our model, we assume that all the sensor nodes are operated at low-duty-cycle mode, the working schedule of each node is periodic and alternates between one *active state* and $L - 1$ *sleep states*.

As the same with most of literature for low-duty-cycle WSNs [1]–[4], we assume *time synchronization is achieved, and each node can transmit its packets at any time according to its neighbors' working schedules while can only receive the packets from its neighbors in active states*. Besides, we also have the following basic assumptions: (1) Each node is aware of the working schedules of all its neighboring nodes within 2 hops, this can be realized via local information exchange between neighboring nodes initially after the network is deployed; (2) We do not consider the packet collision problem here due to the fact that the low-duty-cycle operation inherently reduces the probability of collision to a great extent, which has been experimentally verified in [2]; (3) We regard the broadcasting message transmission as the main energy consumption source.

B. Problem Formulation

Obviously, the broadcasting schedule where all nodes receive the broadcasting message at their scheduled wake-up time slots could bring the shortest broadcasting latency, however, draw much more energy consumption since any one-hop broadcasting is actually realized by a number of unicasts. Nevertheless, we find that the transmission number of the broadcasting message could be further reduced by considering broadcasting spatiotemporal locality, i.e. deferring the receiving time of some nodes. For any sender, here, we define two kinds of receivers: DelayedReceiver and InstantReceiver. In our model, the sender will send the broadcasting message to each InstantReceiver, and also it will send a short Beacon packet that only contains the ID of some InstantReceiver j , saying *Beacon*(j), to each DelayedReceiver. Upon receiving the *Beacon*(j) from the sender, any DelayedReceiver will go to sleep immediately and defer its receiving time by setting a timer to wake up at the *active state* of the InstantReceiver j . Due to assumption (1), actually, the DelayedReceiver can be aware of the working schedule of the InstantReceiver j .

Definition 1 (Forwarding Sequence). For any forwarder i of the broadcasting message, its Forwarding Sequence $S_f(i)$ is defined as a sequence of its receivers sorted based on the scheduled wake-up time, namely

$$S_f(i) = \langle r_1^1, \dots, r_1^{k_1}, \underline{r_1}, r_2^1, \dots, r_2^{k_2}, \underline{r_2}, \dots, r_j^1, \dots, r_j^{k_j}, \underline{r_j} \rangle$$

where r_j^k ($k = 1, \dots, k_j$) and the underlined r_j respectively denote the DelayedReceivers and InstantReceivers of node i . Specifically, the forwarder i will send the short control packet *Beacon*(r_j) to each DelayedReceiver r_j^k and send the broadcasting message to each InstantReceiver r_j .

Definition 2 (Broadcasting Schedule). Given a communication graph $G = (V, E)$ in which V is the set of N nodes including the sink v_0 and all sensing nodes $I = \{v_i | i = 1, \dots, N - 1\}$, and E is the set of all communication edges, the schedule strategy of any sensing

node v_i ($i = 1, \dots, N - 1$) in G , saying $M(v_i)$, can be defined as follows:

$$M(v_i) = (\alpha, \beta) \quad (1)$$

where

$$\alpha \in \{0, 1\}, \quad \beta = \begin{cases} S_f(v_i) & \alpha = 1 \\ NULL & \alpha = 0 \end{cases}$$

In Equ. (1), the binary variable α denotes whether node v_i be a forwarder after receiving the broadcasting message, and if v_i is the forwarder (i.e. $M(v_i).\alpha = 1$), β will denote the Forwarding Sequence $S_f(v_i)$, which represents that once receiving the broadcasting message, node v_i will send the short Beacon packet or the broadcasting message to each node in $S_f(v_i)$ in sequence. Here, *NULL* denotes the omitted item and it is obvious that $M(v_i).\beta = NULL$ for any node v_i with $M(v_i).\alpha = 0$.

Here, a broadcasting schedule \mathbb{M} in the network can be defined as the set of all nodes' schedule strategies:

$$\mathbb{M} = \{M(v_i) | i = 0, \dots, N - 1\} \quad (2)$$

so that 1) I_α is a connected vertices subset in G ; 2) $\bigcup_{v_i \in I_\alpha} M(v_i).\beta = I$; and 3) $M(v_i).\beta \cap M(v_j).\beta = \emptyset$ for any $v_i, v_j \in I_\alpha$ and $i \neq j$, where $I_\alpha = \{v_i | i = 0, \dots, N - 1 \text{ and } M(v_i).\alpha = 1\}$.

In the above definition, assumption (3) still holds since in practice, the energy consumption for transmitting and receiving the short Beacon packet in our model is so small that it can be neglected compared with that for transmitting the broadcasting message, especially for the applications of large message broadcasting such as *code updates*. Here, our objective is to address the following *Latency-optimal Minimum Energy Broadcast (L-MEB) Problem*.

Problem 1 (L-MEB). Given a communication graph $G = (V, E)$, how to find an efficient broadcasting schedule \mathbb{M} to optimize the transmission number of the broadcasting message, i.e. to minimize $\sum_{i=0}^{N-1} M(v_i).\alpha$, while guaranteeing that the broadcasting latency is minimized.

Definition 3 (Coverage Set). The Coverage Set of any Sender-InstantReceiver pair (v_s, v_r) at time slot t ($0 \leq t \leq L - 1$), saying $CS(v_s, v_r, t)$, is defined as follows: if $t < T_s(v_r)$, $CS(v_s, v_r, t) = \{x \in N(v_s) - \{v_0\} | t < T_s(x) \leq T_s(v_r)\}$; otherwise, $CS(v_s, v_r, t) = \{x \in N(v_s) - \{v_0\} | t < T_s(x) \leq L - 1 \text{ or } 0 \leq T_s(x) \leq T_s(v_r)\}$ in which v_0 denotes the sink node, $N(v_i)$ and $T_s(v_i)$ denote the neighboring nodes set and the scheduled wake-up time slot of node v_i respectively.

Observation 1. Given a communication graph $G = (V, E)$, if any node v_s decides to send the broadcasting message to its neighbor v_r , then an efficient broadcasting schedule

must make sure that when being received by v_r , the broadcasting message also has been received by all the nodes in $CS(v_s, v_r, T_c(v_s))$ where $T_c(v_s)$ denotes the time slot that the uncovered node v_s receives the broadcasting message.

In order to better exhibit the spatiotemporal characteristic of broadcasting, we can transform the original topology graph into the Spatiotemporal Relationship Graph (SRG), in which each edge represents one transmission and its ending vertex represents the resulting coverage set after this transmission, by the following Spatiotemporal Relationship Graph Construction Algorithm (SRGC-A): Initially, SRG only contains a vertex $\{v_0\}$. Starting with considering the sink v_0 as the sender, we respectively regard each neighbor i of the sink as the InstantReceiver and insert a directed edge from the vertex $\{v_0\}$ to the newly added vertex $CS(v_0, i, T_s(v_0))$ ($i \in N(v_0)$). For each newly added vertex, saying CS_{new} , we in turn select each node $j \in CS_{new}$ as the sender, then search all the vertices in SRG to check whether the vertex $CS(j, i, T_s(k))$ ($i \in N(j) - \{v_0\}$), where node k denotes the InstantReceiver in CS_{new} , has existed. If so, we just insert a directed edge between the vertex CS_{new} and this existing vertex; otherwise, we add the vertex $CS(j, i, T_s(k))$, as well as an edge connecting $CS(j, i, T_s(k))$ with CS_{new} , into SRG. The above process repeats until no further vertex insertion to SRG is possible. Finally, we attach each vertex in SRG with the time slot of the InstantReceiver in its corresponding coverage set, and for each edge representing one transmission, we also mark it with its corresponding Sender-InstantReceiver pair. Here, we can easily show that the worst-case time complexity of SRGC-A is $O(N^2 d_{max}^6)$, where d_{max} denotes the maximum node degree in the original topology graph.

Letting $D(T)$ denote the latency of any tree T , we define that $D(T) = \max_{t \in leaf(T)} \{D_T(r, t)\}$ where r denotes the root of the tree, $D_T(i, j)$ and $leaf(T)$ denote the E2E latency from vertex i to vertex j on T and the set of the leaves of T respectively, and further, we define the *Latency-optimal Group Steiner Tree (L-GST) Problem* as follows.

Problem 2 (L-GST). Given a directed graph $G' = (V', E')$ with weight $w_{e'} = 1$ for each edge $e' \in E'$ and a family of subsets (groups) of vertices $f = \{g_1, g_2, \dots, g_k\}$ ($g_i \subseteq V'$), how to find a minimum weight subtree $T_{opt} = (V_T \subseteq V', E_T \subseteq E')$ rooted from a specified vertex such that: (1) $V_T \cap g_i \neq \phi$ for all $i \in \{1, \dots, k\}$; (2) the latency of T_{opt} is minimal over all the subtrees that satisfy constraint (1).

In SRG, we can partition all the vertices into N groups according to the common members in their coverage sets. For any node i in the original topology graph, we classify all the SRG vertices whose coverage sets containing node i into a group. Obviously, one broadcasting schedule can be implicitly represented by a subtree of SRG which is rooted from the vertex $\{v_0\}$ and connects at least one vertex in each

group of SRG. Therefore, L-MEB problem on the original topology graph can be transformed into L-GST problem on its corresponding SRG. Also, we define the Minimum Latency Path Tree (MLPT) in any graph G as the spanning subset of G which consists of the minimum latency paths from the root to all the vertices in G , and we can easily find that under our model, the latency of MLPT in the original topology graph is the optimal broadcasting latency. Here, we can easily show the NP-hardness of L-MEB problem by reduction from the Set Cover Problem, the detailed proof process is omitted due to space limitation.

III. APPROXIMATION ALGORITHM

In order to solve the aforementioned problem, in this section, we propose an efficient approximation solution which consists of the following steps.

A. LGT Construction

Given any original topology graph G and its corresponding SRG G' , we can derive the optimal broadcasting latency $D_{opt}(G)$ by figuring out the latency of MLPT in G . In addition, we can further simplify SRG by merely retaining the vertices in G' whose minimum root-to-vertex latencies in G' are not more than $D_{opt}(G)$. This is because our expected subtree of G' which represents the latency-optimal broadcasting schedule, will absolutely not include any vertex whose minimum root-to-vertex latency in G' is more than the optimal broadcasting latency. Thus, our target problem will further turn to be L-GST problem on the simplified SRG.

We use $OPT_{GST}(T)$ and $OPT_{L-GST}(G)$ to denote the cost of the optimal solution for Group Steiner Tree (GST) Problem on any tree T and that for L-GST problem on any graph G , respectively, and the following conclusion holds.

Theorem 1. For any latency-optimal spanning subtree T^+ of the simplified SRG G^+ , we have $OPT_{GST}(T^+) \leq h(T^+) \cdot OPT_{L-GST}(G^+)$, where $h(T^+)$ denotes the height of tree T^+ and is bounded by a constant $\xi = \frac{R-L}{r_c}$, where r_c denotes the communication range of each node.

Proof: The proof is omitted due to space limitation. ■

According to Theorem 1, we are expected to find a latency-optimal spanning subtree of the simplified SRG G^+ , which we call the Latency-optimality Guaranteed Tree (LGT), to provide the performance guarantee. Obviously, the MLPT in G^+ must be a latency-optimal spanning subtree of G^+ and therefore can be directly taken as the LGT.

B. Edge Selection on LGT

Seeing from above, actually, we can approximate our problem as the GST problem on LGT which has guaranteed the optimality of broadcasting latency. In [7], authors propose an efficient method, which we call the Randomized-rounding based Edge Selection Algorithm (RES-A), to address the GST problem on tree. In general, RES-A [7]

first converts the input tree into an equivalent binary tree where each group is a subset of its leaves and groups are pairwise disjoint, then transforms the GST problem on the transformed binary tree into a 0-1 Integer Programming and adopts a randomized-rounding based approach to get the Selected Edge Set in which all the selected edges constitute the solution, herein, the randomized-rounding will be repeated for $\lceil \eta \cdot \log(N-1) \cdot \log \max_{1 \leq i \leq N-1} |g_i| \rceil$ rounds, where η is a constant. Finally, the solution on the transformed binary tree will be reduced to that on the original input tree by removing all the zero-weight edges from the Selected Edge Set. However, RES-A only gives a probabilistic solution. Based on RES-A, here, we will propose a Deterministic Randomized-rounding based Edge Selection Algorithm (DRES-A) by the following edge compensation operation: If the root is not connected to some group g after running RES-A on LGT, specifically, we will establish the minimum weight path from the root to group g and then add the edges on this path which have not been selected by RES-A into the Selected Edge Set.

C. Broadcasting Schedule Construction

By adopting the above-mentioned solution, we can approximately obtain the minimum weight Group Steiner Tree on LGT that consists of the edges in Selected Edge Set, saying $T^G = (V^G, E^G)$, which implicitly represents an energy efficient latency-optimal broadcasting schedule. As the randomized approach is adopted in the building of T^G , the resulting broadcasting schedule could incur redundant transmissions as well as unnecessary collisions. Next, we will introduce how to transform T^G into the corresponding broadcasting schedule as defined in Definition 2, which essentially avoids the redundant transmissions and unnecessary collisions.

For any vertex $v_i^G \in V^G$, $T_s(v_i^G)$ is used to denote the scheduled wake-up time slot of the InstantReceiver in v_i^G (namely $T_s(v_i^G) = T_s(v_j)$ where v_j is the InstantReceiver of vertex v_i^G). For any edge $e_i^G = (v_s^G, v_r^G)$ in E^G which represents one transmission, we use a four-tuple $(S(e_i^G), t_{S(e_i^G)}, T_c(S(e_i^G)), R(e_i^G))$ to characterize it, in which $S(e_i^G)$ and $R(e_i^G)$ respectively denote the Sender and the InstantReceiver of the transmission; $t_{S(e_i^G)}$ denotes the time when the sender receives the broadcasting message before the transmission e_i^G , and $T_c(S(e_i^G))$, which equals to $T_s(v_s^G)$, denotes the corresponding time slot of $t_{S(e_i^G)}$. For any sensor node v_i , we use $t_{v_i}^{min}$ to denote the time when node v_i is covered for the first time in the schedule T^G , specifically,

$$t_{v_i}^{min} = \min_{k \in V^G(v_i)} D_{T^G}(r^G, k) \quad (3)$$

where r^G is the root of T^G , $V^G(v_i)$ is the subset of V^G consisting of the vertices of which coverage sets contain node v_i . Further, for any forwarder v_j , we define $T_c^{min}(v_j)$

as follows:

$$T_c^{min}(v_j) = T_s(\arg \min_{k \in V_s^G(v_j)} D_{T^G}(r^G, k)) \quad (4)$$

where $V_s^G(v_j)$ is the subset of $V^G(v_j)$ consisting of the vertices which have at least one output edge with the sender v_j in T^G . Here, we use a ring to characterize one working schedule period, i.e. time slots from 0 to L-1 are distributed in the ring according to the clockwise sequence.

Here, we propose a Broadcasting Schedule Construction Algorithm (BSC-A), which includes the following two steps:

Schedule Initialization: For any node v_i , its schedule strategy $M(v_i)$ can be initially generated from T^G as follows: If there is no edge where the sender is v_i in E^G , we set $M(v_i).\alpha = 0$ and $M(v_i).\beta = NULL$. If there exists at least one edge indicating the sender is v_i in E^G , we set $M(v_i).\alpha = 1$ and $M(v_i).\beta$ which is initially null can be built by the following way: For any edge e_i^G in the set of edges where the sender is v_i , we check that whether $CS(v_i, R(e_i^G), T_c(S(e_i^G))) \subseteq CS(v_i, R(e_i^G), T_c^{min}(v_i))$, if yes, we add node $R(e_i^G)$ into $M(v_i).\beta$ if it is not in $M(v_i).\beta$ and mark it as the InstantReceiver; otherwise, node v_i' will be added into $M(v_i).\beta$ if it is not in $M(v_i).\beta$ and be marked as the InstantReceiver, where v_i' is the neighboring node of v_i whose scheduled wake-up time slot is the furthest away from the time slot $T_c^{min}(v_i)$ in the wake-up schedule ring along with the clockwise direction. Then, we sort $M(v_i).\beta$ as $\langle \beta_1, \beta_2, \dots, \beta_{m(v_i)} \rangle$ according to the clockwise sequence of their scheduled time slots in the wake-up schedule ring with starting from the time slot $T_c^{min}(v_i)$. Afterwards, we add all the nodes in set $CS(v_i, \beta_{m(v_i)}, T_c^{min}(v_i)) - M(v_i).\beta$ into $M(v_i).\beta$ and mark them as the DelayedReceivers, and then we reorder $M(v_i).\beta$ according to the clockwise sequence of their scheduled time slots in the wake-up schedule ring with starting from the time slot $T_c^{min}(v_i)$.

Schedule Adjustment: For each non-sink node v_i , we first find the edge $e_i^G = (v_s^G, v_r^G)$ in E^G such that $v_i \in v_r^G$ and $D_{T^G}(r^G, v_r^G) = t_{v_i}^{min}$, and node $S(e_i^G)$ is thus selected to be the candidate sender for v_i . Then, we check the Forwarding Sequence of each forwarder v_j where $v_j \neq S(e_i^G)$, if $v_i \in M(v_j).\beta$, then 1) if v_i is not the InstantReceiver, to remove v_i from $M(v_j).\beta$; 2) otherwise, to replace v_i with the previous node of v_i in $M(v_j).\beta$ to be the InstantReceiver and then remove v_i from it, particularly, if the previous node of v_i is also the InstantReceiver or v_i is the first node in $M(v_j).\beta$, just to remove v_i from $M(v_j).\beta$. After the above process, if the new resulting Forwarding Sequence of any forwarder v_j is empty, we will set $M(v_j).\alpha = 0$ and $M(v_j).\beta = NULL$.

Proposition 1. *The broadcasting schedule resulted from BSC-A will keep the latency-optimality and must be no worse than the schedule that represented by T^G in terms of the number of transmissions.*

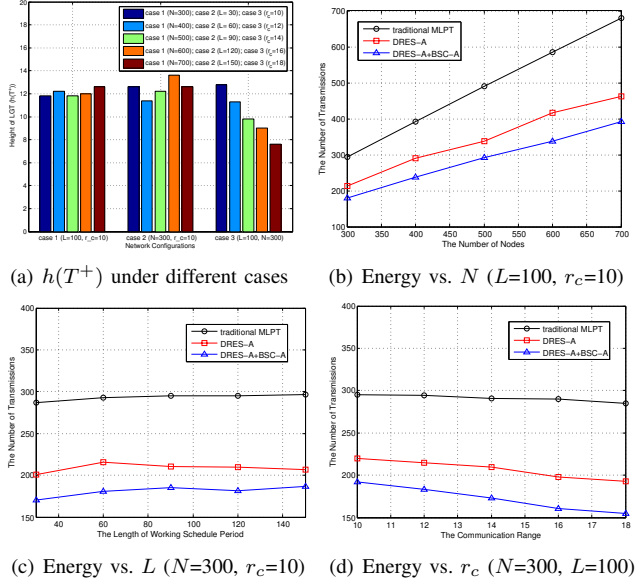


Figure 1: The performance under different configurations.

Theorem 2. When $\eta \geq 64$, the approximation ratio of our solution is $O(\log N \cdot \log d_{max})$.

Proof: The proof is omitted due to space limitation. ■

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our solution via simulations. Here, we assume $R = 50$ and once the length of working schedule period L is fixed, each node independently and randomly chooses a time slot to wake up in one period of L time slots and then repeat this working schedule. All the results are obtained by averaging results of 10 experiments.

First, we evaluate the value of $h(T^+)$ where T^+ denotes LGT, namely the MLPT on the simplified SRG. We respectively consider the following three cases of network configurations: ($L=100, r_c=10$), ($N=300, r_c=10$) and ($L=100, N=300$). As shown in Fig. 1(a), the value of $h(T^+)$ almost keeps stable, namely around 12, as the number of nodes or the length of working schedule period increases. However, $h(T^+)$ drops as the communication range of each node increases, which means it is only related to the communication range of each node on the condition that R is fixed. According to the simulation result depicted in Fig. 1(a), we can obviously find that the value of $h(T^+)$ is always a small constant without being related with N and is far less than its theoretical upper bound $\frac{R \cdot L}{r_c}$ under whatever the network configuration, thus it can be approximately neglected in the computation of the approximation ratio.

Next, we proceed to evaluate the performance of our proposed approximation solution by comparing it with the traditional MLPT-based latency-optimal strategy, in which

no deferring policy is employed and the sink broadcasts the message directly along with the MLPT of the original topology graph. As shown in Fig. 1(b)-(d), our solution using both DRES-A and BSC-A significantly reduces the total transmission number of the broadcasting message compared with the traditional MLPT-based approach under various network configurations, and achieves around 10%-25% improvement over the solution where only DRES-A is adopted, i.e. $|E^G|$, which shows the high-efficiency of our proposed BSC-A on the reduction of redundant transmissions. Besides, seeing from Fig. 1(b)-(d), it is clear that the network density and the transmission power affect the performance of our solution to a greater extent compared with the duty cycle given that R is fixed, and specifically, our solution would perform better as the network density or the transmission power rises.

V. CONCLUSION

In this paper, we consider how to utilize broadcasting spatiotemporal locality to address the broadcast scheduling problem in low-duty-cycle WSNs. We first transform our target problem into the Latency-optimal Group Steiner Tree Problem on the Spatiotemporal Relationship Graph, which is NP-hard, and then approximately solve this problem by using a deterministic randomized-rounding based method. Also, an efficient algorithm is proposed to further avoid redundant transmissions and unnecessary collisions. Finally, the high-efficiency of our solution has been evaluated through both theoretical analysis and simulations.

ACKNOWLEDGMENT

The work is partly supported by China 973 project (2012CB316200) and China NSF grants (61133006, 61021062, 60903179).

REFERENCES

- [1] S. Guo, Y. Gu, B. Jiang, and T. He. *Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links*. In MobiCom, 2009.
- [2] F. Wang and J.C. Liu. *Duty-cycle-aware broadcast in wireless sensor networks*. In INFOCOM, pages 468-476, 2009.
- [3] S. Guo, S.M. Kim, T. Zhu, Y. Gu, T. He. *Correlated flooding in low-duty-cycle wireless sensor networks*. In ICNP, 2011.
- [4] X.L. Jiao, W. Lou, J.C. Ma, J.N. Cao, X.D. Wang, X.M. Zhou. *Duty-cycle-aware minimum latency broadcast scheduling in multi-hop wireless networks*. In ICDCS, 2010.
- [5] Z.J. Li, M. Li, J.L. Liu, S.J. Tang. *Understanding the flooding in low-duty-cycle wireless sensor networks*. In ICPP, 2011.
- [6] Y.J. Sun, O. Gurewitz, S. Du, L. Tang, and D.B. Johnson. *ADB: An efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks*. In SenSys, pages 43-56, 2009.
- [7] N. Garg, G. Konjevod, R. Ravi. *A polylogarithmic approximation algorithm for the group steiner tree problem*. In SODA, 2000.