JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 9, No. 2, June 2015, pp. 83-97

A Holistic Approach to Optimizing the Lifetime of IEEE 802.15.4/ZigBee Networks with a Deterministic Guarantee of Real-Time Flows

Kang-Wook Kim
Department of Computer Science and Engineering, Seoul National University, Seoul, Korea
kwkim@rubis.snu.ac.kr

Myung-Gon Park
SAP Labs Korea Inc., Seoul, Korea
myunggon.park@sap.com

Junghee Han
Department of Telecommunication and Computer Engineering, Korea Aerospace University, Goyang, Korea
junghee@kau.ac.kr

Chang-Gun Lee*
Department of Computer Science and Engineering, Seoul National University, Seoul, Korea
cglee@snu.ac.kr

Abstract
IEEE 802.15.4 is a global standard designed for emerging applications in low-rate wireless personal area networks (LR-WPANs). The standard provides beneficial features, such as a beacon-enabled mode and guaranteed time slots for real-time data delivery. However, how to optimally operate those features is still an open issue. For the optimal operation of the features, this paper proposes a holistic optimization method that jointly optimizes three cross-related problems: cluster-tree construction, nodes' power configuration, and duty-cycle scheduling. Our holistic optimization method provides a solution for those problems so that all the real-time packets can be delivered within their deadlines in the most energy-efficient way. Our simulation study shows that compared to existing methods, our holistic optimization can guarantee the on-time delivery of all real-time packets while significantly saving energy, consequently, significantly increasing the lifetime of the network. Furthermore, we show that our holistic optimization can be extended to take advantage of the spatial reuse of a radio frequency resource among long distance nodes and, hence, significantly increase the entire network capacity.

Category: Ubiquitous computing

Keywords: Holistic optimization; Real-time; Wireless sensor networks

Open Access http://dx.doi.org/10.5626/JCSE.2015.9.2.83 http://jcse.kiise.org

Received 13 May 2015; **Accepted** 25 May 2015
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

I. INTRODUCTION

IEEE 802.15.4 is a global standard for emerging applications in low-rate wireless sensor networks (LR-WPANs). Its targeting applications include health monitoring, disaster detection/reporting, target tracking, and factory automation. In those applications, a number of real-time data flows are ongoing and their time-sensitive packets need to be delivered on time. This real-time guarantee must be provided in an energy-efficient manner in order to maximize the network lifetime. Such an energy-efficient real-time guarantee is important because the replacement of batteries of sensor nodes is not only very cumbersome but also practically impossible in some applications such as a densely deployed large-scale sensor network.

For the energy efficiency and the real-time guarantee, the beacon-enabled mode of IEEE 802.15.4 provides beneficial features, such as synchronized operations with small duty-cycles and guaranteed time slots for collision-free transmissions. However, how to optimally utilize those features is still not completely understood. In this paper, we propose a holistic approach to optimally configuring the IEEE 802.15.4/ZigBee cluster-tree network jointly addressing the three cross-related problems: logical cluster-tree construction, power configuration for nodes, and duty-cycle scheduling of clusters. Let us consider a sensor network that has six ZigBee nodes and two real-time data flows as shown in Fig. 1. In order to guarantee the end-to-end deadline of every data packet of the given flows, we first have to construct the logical cluster-tree so that packets can be routed along the tree structure. This problem of logical cluster-tree construction is directly related to the power configuration problem of all nodes since the powers of child and parent nodes need to be properly configured to ensure that their radio frequency (RF) signals are bi-directionally reachable. Once a cluster-tree has been created, we also have to determine the duty-cycle scheduling of all the clusters so that each node in a cluster can send a packet using its dedicated guaranteed time slot (GTS) within the cluster's active period called a superframe duration (SD), which periodically occurs at every beacon interval (BI). This duty-cycle scheduling problem should also determine the lengths of SDs and BIs of all the clusters and allocations of GTSs to all the nodes. Thus, the duty-cycle scheduling problem and cluster-tree construction are inter-dependent. Furthermore, the resulting cluster-tree and duty-cycle scheduling affect the end-to-end delay of all the real-time flows. In addition, the resulting power-configuration and duty-cycle scheduling affect the overall energy consumption and, in turn, the lifetime of the network. Therefore, we need to jointly address the three problems together in

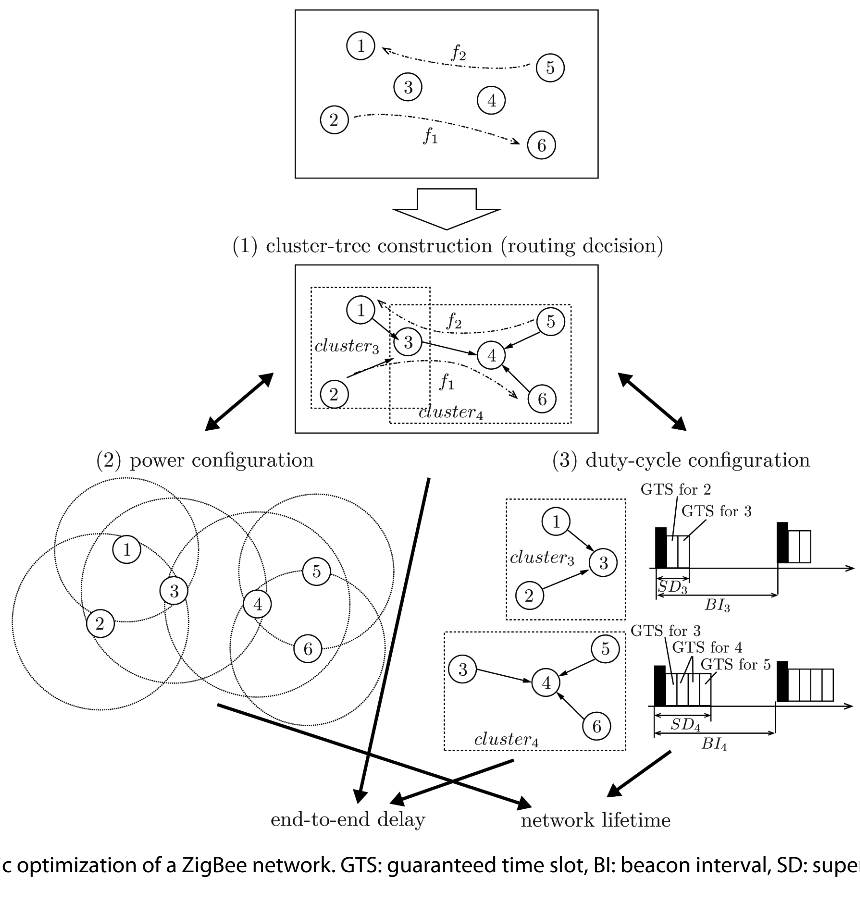

Fig. 1. A holistic optimization of a ZigBee network. GTS: guaranteed time slot, BI: beacon interval, SD: superframe duration.

order to guarantee that all the real-time flows occur in the most energy efficient way; this is what we propose in this paper. Existing research has only tried to solve part of these three related problems.

For example, Han only addresses the duty-cycle scheduling problem assuming that the logical cluster-tree and power configurations are already given [1]. Ergen and Varaiya [2] presents an energy efficient routing method with a delay guarantee for wireless sensor networks assuming that duty-cycle scheduling and power configurations are already given [2]. However, no existing research proposes a holistic solution to optimally solve all three related problems together.

In addition to the above three related problems, for a large-scale network, we encounter one more problem of how to optimally reuse the RF so that as many nodes as possible can simultaneously transmit packets without collisions if they are geographically far apart. This is called *spatial reuse of RF resource*. For the optimal spatial reuse of RF resource, we have extended our proposed holistic approach to solve the spatial reuse problem as well as the above three problems.

The rest of this paper is organized as follows: the next section surveys the related work. Then, Section III overviews the IEEE 802.15.4/ZigBee standard and formally defines the problem to be addressed in this paper. Section IV proposes our holistic optimization framework to jointly solve the above three cross-related problems. Section V extends our holistic optimization framework to additionally address the problem of spatial reuse of the RF resource. Section VI presents our experiments. Finally, Section VII concludes this paper.

II. RELATED WORK

Recently, the issues for real-time data delivery have been extensively addressed in different settings of wireless sensor networks. In broad terms, real-time medium access control (MAC) protocols, real-time routing protocols, and real-time MAC/routing cross-layer protocols have been proposed. For example, the implicit EDF [3] is a hard real-time MAC protocol that provides a collision-free real-time packet scheduling scheme that exploits the periodic nature of real-time data flows. The dual-mode real-time MAC protocol [4] provides both bounded worst-case delays for real-time packets and good average delays for best-effort packets by switching between protected and unprotected modes. On the other hand, real-time power-aware routing (RPAR) [5] is an example of real-time routing protocols that meet the application specified delay bound requirements at a low energy cost by dynamically configuring transmission power and routing decisions. Ergen and Varaiya [2] propose another routing method that finds the optimal routing paths for real-time data flows in a way that maximizes the network lifetime. The method uses a linear programming formulation to find the optimal routing solution assuming that the power configuration and the network topology are given as inputs. SPEED [6] is an example of MAC/routing cross-layer protocols that support real-time packet deliveries. SPEED is designed to provide soft end-to-end deadline guarantees by enforcing uniform packet delivery speeds over the entire network through feedback control in the MAC layer and geographic packet forwarding in the routing layer. MMSPEED [7] extends the SPEED protocol by providing multiple speeds in order to provide service differentiations for different classes of real-time flows. All the above methods, however, are designed for sensor network settings that different significantly to the IEEE 802.15.4/ZigBee standard. Thus, they cannot be applied to build the optimal IEEE 802.15.4/ZigBee network. More importantly, even in this broad scope, there is no existing work that holistically and simultaneously optimizes topology construction, power configuration, and packet scheduling.

In the specific scope of the IEEE 802.15.4/ZigBee standard [8], Koubaa et al. [9] propose algorithms to schedule SDs of all the clusters in a collision-free way for the given values of SDs and BIs for all the clusters. However, they do not address how to optimally determine the SD and BI values for all clusters. In [1], Han addresses the optimal duty-cycle scheduling problem, that is, optimally finding the SD, BI values and also GTS allocations for guaranteeing all the real-time flows while maximizing the network lifetime. Han's approach, however, only addresses the duty-cycle scheduling problem assuming that the cluster-tree topology and power configurations are given as inputs. None of the IEEE 802.15.4/ZigBee related work addresses the holistic optimization problem that considers cluster-tree construction, power configuration, and duty-cycle scheduling, all together.

The IEEE 802.15.4/ZigBee standard provides a beacon-enabled mode for the energy-efficient delivery of real-time packets. In the beacon-enabled mode, all the ZigBee nodes first need to form a logical cluster-tree network. As an example, Fig. 2(a) shows physical deployments of ZigBee nodes, denoted by N_1, N_2, ⋯, N_8. The nodes form a logical cluster-tree as in Fig. 2(b) where N_1, N_2, and N_3 form a cluster with N_3 as the cluster head, N_7 and N_8 form another cluster with the head of N_7, and N_3, N_4, N_5, N_6, and N_7 form an upper level cluster with the head of N_4. This tree formation is necessary for the routing of packets and scheduling of packets. First, the routing path of a packet from a source node N_s to a destination node N_d is simply determined along the structure of the cluster-tree. For example, the routing path from N_5 to N_2 is $N_5 \rightarrow N_4 \rightarrow N_3 \rightarrow N_2$. Second, the packet transmission times are scheduled in a cluster-based collision-free manner. More specifically, each cluster periodically has its dedicated active duration called *SD* which is not related to the SDs of other clusters as shown in Fig.

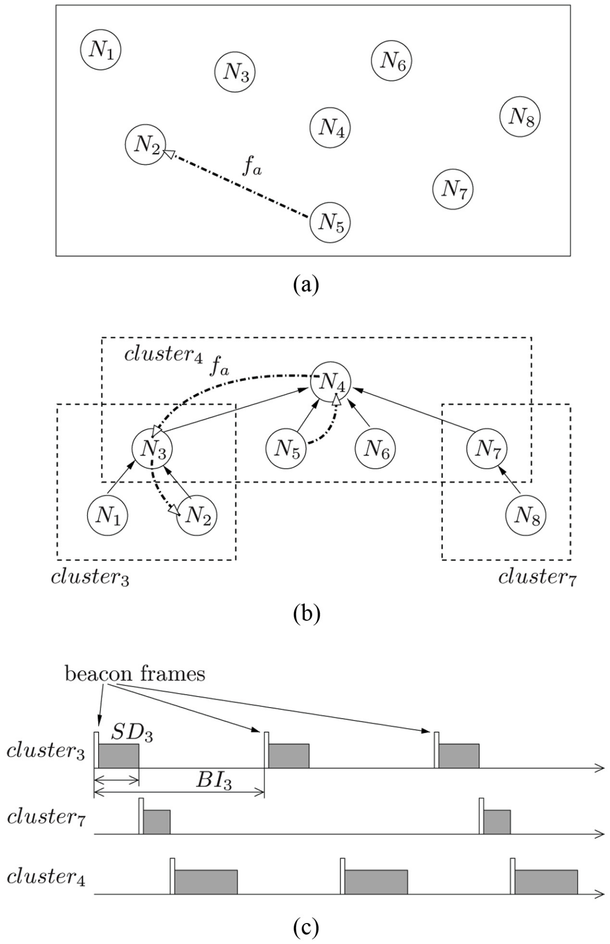

Fig. 2. An example ZigBee network. (a) Physical placement of nodes, (b) logical cluster tree view, and (c) clusters' SD scheduling. BI: beacon interval, SD: superframe duration.

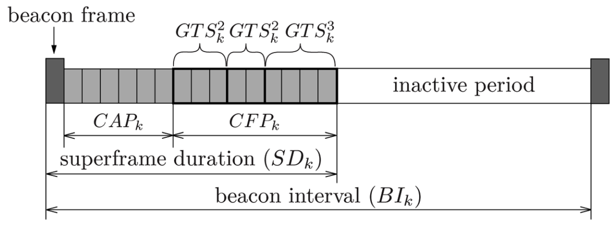

Fig. 3. Beacon interval (BI) and superframe duration (SD) concepts. GTS: guaranteed time slot, CAP: contention access period, CFP: contention-free period.

2(c). Only within the cluster's *SD*, all the nodes of the cluster wake up and communicate with each other. For this, the head of each cluster sends a beacon frame at every beacon period, called *BI*, as shown in Fig. 2(c). With the beacon frame, all the nodes belonging to the cluster synchronize. Each beacon period BI_k of a cluster k is composed of the cluster's active period SD_k and its inactive period as shown in Fig. 3. The SD_k contains a contention access period (CAP_k), in which nodes compete in a slotted carrier sense multiple access with collision avoidance (CSMA/CA) manner for non-real-time packets, and a contention-free period (CFP_k), in which nodes transmit their real-time packets with their dedicated guaranteed time slots (GTS_ks).

As such, the IEEE 802.15.4/ZigBee standard provides baseline features for routing and scheduling real-time packets. However, for the proper operation of the ZigBee network, we still have to find the optimal configuration of all the operating parameters for

- cluster-tree construction,
- BIs, SDs for all the clusters and GTS allocations for the member nodes, and
- power configuration of all the nodes to enable the formation of the desired cluster-tree.

In order to formally define this holistic optimization problem, we provide the following definitions and notations:

Cluster-tree construction related definitions: A cluster-tree is represented by the parent-child relation ω_{ij} for every pair of two nodes N_i and N_j, which is defined as follows:

$$\omega_{ij} = \begin{cases} 1 & \text{if } N_i \text{ is the parent of } N_j \\ 0 & \text{otherwise} \end{cases}.$$

In the example of Fig. 2(b), ω_{31} and ω_{45} are 1 while ω_{13}, ω_{25} and ω_{52} are 0. With this ω_{ij} notation, a node N_i and all its direct child nodes N_js for which $\omega_{ij} = 1$ form a cluster. For the cluster, N_i is called the head while all its child nodes N_js are called members. Thus, we use the cluster head's id as the cluster's id. For the example of Fig. 2(b), we have three clusters: $cluster_3$ with head N_3, $cluster_7$ with head N_7, and $cluster_4$ with head N_4.

Duty-cycle scheduling related definitions: As mentioned in Fig. 2(c), each $cluster_k$'s activity window is characterized by the superframe duration SD_k and the beacon period BI_k. The length of BI_k and SD_k are determined by two parameters, *beacon order* (BO) and *superframe order* (SO), respectively, as follows:

$$\begin{gathered} BI_k = aBaseSuperframeDuration \cdot 2^{BO_k} \\ SD_k = aBaseSuperframeDuration \cdot 2^{SO_k}, \\ \text{for } 0 \le SO_k \le BO_k \le 14, \end{gathered} \tag{1}$$

where *aBaseSuperframeDuration* denotes the minimum length of the superframe, corresponding to $SO_k = 0$. This duration is fixed to 960 symbols (a symbol corresponds to 4 bits) by the standard. This value corresponds to the fixed duration of 15.36 ms, assuming a 250 kbps in the 2.4 GHz frequency band. We call this fixed *aBaseSuperframeDuration* a slot. Thus, the problem of determining the lengths of BI_k and SD_k for every $cluster_k$ is to determine the integer values of BO_k and SO_k in the range from 0 to 14.

The head node and all the member nodes of $cluster_k$ are allowed to transmit their packets only for the duration of SD_k. The slots in a part of SD_k, denoted by CAP_k in Fig. 3, are accessed by the cluster's nodes in a contention based manner for non-real-time packets. For the non-real-time packets, it is required to provide the minimum capacity, denoted by $MIN_{nonRealTime}$, that is,

$$\frac{CAP_k}{BI_k} \geq MIN_{nonRealTime},$$

where CAP_k is an integer multiple of a slot. In this paper, we consider only real-time packets and, hence, assume that $MIN_{nonRealTime} = 0$. However, our proposed optimization framework works for any value of $MIN_{nonRealTime}$.

For the nodes that deliver real-time data flows, we have to allocate GTS_ks (guarantee time slots) as in Fig. 3. The length of a GTS allocated to a node N_i in $cluster_k$ is denoted by GTS_k^i, which is an integer multiple of a slot.

In case that N_i is the $cluster_k$'s head, that is, $N_i = N_k$, its GTS, that is, GTS_k^k is used for N_k to transmit real-time data to its member nodes N_j for all j where $\omega_{kj} = 1$. The portion of GTS_k^k used for transmitting data from N_k to N_j is denoted by GTS_k^{kj}. In other words, $GTS_k^k = \sum_{\forall j, \omega_{kj}=1} GTS_k^{kj}$.

In case that N_i is a member of $cluster_k$, that is, $N_i \neq N_k$, its GTS, that is, GTS_k^i is used for N_i to transmit real-time data only to its head N_k. That is, if we use the same notational meaning GTS_k^{ik}, that is, the portion used for N_i to transmit real-time data to N_k, we note that $GTS_k^i = GTS_k^{ik}$.

Nodes' power related definitions: For the successful transmission in between two nodes, their powers need to be properly configured. For this, we control the transmission power denoted by PW_i of each node N_i assuming that every node in a receive mode uses the same power PW_{recv}.

With these definitions, our problem is described as follows.

Problem Description: We are given a set of sensor nodes $V = \{N_1, N_2, \cdots, N_m\}$ whose locations are pre-fixed. Thus, we know the geometric distance between every pair of two nodes, N_i and N_j. The distance is denoted by d_{ij}.

We are also given a set Γ of n periodic real-time data flows $f_1, f_2, \cdots, f_n$, i.e.,

$$\Gamma = \{f_1, f_2, \cdots, f_n\}.$$

A periodic flow f_a is characterized by a 5-tuple as follows.

$$f_a = (src_a, dst_a, size_a, p_a, D_a),$$

where src_a, dst_a, $size_a$, p_a, and D_a are the source, destination, packet size (bits), period, and end-to-end deadline, respectively.

For these given set V of nodes and set Γ of real-time flows, our problem is to find all the following three domain parameters:

- cluster-tree construction: ω_{ij} for every pair of N_i and N_j,
- duty-cycle scheduling: BI_k and SD_k for every $cluster_k$ and GTS_k^i for each node N_i of $cluster_k$, and
- nodes' power configuration: PW_i for every node N_i,

This combination makes it possible to maximize the network lifetime can be maximized while guaranteeing the delivery of all the packets of each flow f_a within their end-to-end deadlines. In the next section, we address this problem assuming that only one node in the entire network can transmit a packet at the same time to avoid collisions—*no spatial reuse of RF resource*. Then, Section V relaxes this assumption to allow more than one node to transmit packets at the same time as long as they are far apart and do not cause any collision—*spatial reuse of RF resource*.

IV. PROPOSED HOLISTIC OPTIMIZATION FRAMEWORK WITH NO SPATIAL REUSE OF RF RESOURCE

We explain our optimization objective and constraints in Sections IV-A and IV-B, respectively. Then, in Section IV-C, we explain our genetic algorithm based optimization process.

A. Formulation of the Objective Function for Our Optimization

The objective of our optimization problem is to maximize the lifetime of the network. The network lifetime is the time until the most energy consuming node dies. As a measure of the time, let us consider the 'long-run average power' $avgPW(N_i)$ of a node N_i. With notations introduced in Section III, $avgPW(N_i)$ can be formulated as follows:

$$avgPW(N_i) = \sum_{j=1}^{m} \omega_{ij}\left(\frac{PW_i \cdot GTS_i^{ij} + PW_{recv} \cdot GTS_i^{ji}}{BI_i}\right) + \sum_{j=1}^{m} \omega_{ji}\left(\frac{PW_i \cdot GTS_j^{ij} + PW_{recv} \cdot GTS_j^{ji}}{BI_j}\right). \quad (2)$$

This formula consists of two parts. The first part, i.e., $\sum_{j=1}^{m} \omega_{ij}\left(\frac{PW_i \cdot GTS_i^{ij} + PW_{recv} \cdot GTS_i^{ji}}{BI_i}\right)$, represents the average power for the case when N_i works as the cluster head of

$cluster_i$. For this case, N_i consumes the energy amount of $\sum_{j=1}^{m} \omega_{ij}(PW_i \cdot GTS_i^{ij} + PW_{recv} \cdot GTS_i^{ji})$ for sending to and receiving from its child nodes N_j at every its beacon interval BI_i. Thus, the average power is given as in the first part of Eq. (2).

On the other hand, the second part, i.e., $\sum_{j=1}^{m} \omega_{ji} \left(\frac{PW_i \cdot GTS_j^{ij} + PW_{recv} \cdot GTS_j^{ji}}{BI_j}\right)$, represents the average power for the case when N_i works as a member node of a cluster, say $cluster_j$, that is, solely for the node N_j for which $\omega_{ji} = 1$. For this case, N_i consumes the energy amount of $\sum_{j=1}^{m} \omega_{ji} (PW_i \cdot GTS_j^{ij} + PW_{recv} \cdot GTS_j^{ji})$ for sending to and receiving from its head node N_j at every N_j's beacon interval BI_j. Thus, the average power is given as in the second part of Eq. (2).

With this formulation of $avgPW(N_i)$, the network lifetime is limited by the node that consumes the energy the most, that is, for the node whose long-run average power $avgPW(N_i)$ is the largest. Thus, our objective is to minimize the maximum $avgPW(N_i)$ for all $i = 1, 2, \cdots, m$, that is,

$$\text{Minimize} \max_{1 \le i \le m} avgPW(N_i). \quad (3)$$

B. Formulation of the Constraints for Our Optimization

The above Min-Max problem needs to be solved under a number of constraints described in the following.

Valid tree construction: We use a set of ω_{ij}s, $1 \le i \le m$, $1 \le j \le m$ to represent a tree. For the set to represent a valid tree, ω_{ij}s need to have certain properties. Firstly, every node N_i has at most one parent node. This property can be formulated as the following constraint:

$$\sum_{j=1}^{m} \omega_{ji} \le 1,\ \forall i \in \{1, 2, .., m\}. \quad (4)$$

Secondly, if we add up the numbers of child nodes for all the nodes, it should always be $m - 1$. This property can be formulated as the following constraint:

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \omega_{ij} = m-1 \quad (5)$$

Thirdly, we also have the following obvious constraint, since N_i is not the parent of itself.

$$\omega_{ii} = 0,\ \forall i \in \{1, 2, .., m\}. \quad (6)$$

Finally, if N_i is the parent of N_j, then the other way around is not true. That is,

$$\omega_{ij} + \omega_{ji} \le 1,\ \forall i, j \in \{1, 2, .., m\}. \quad (7)$$

Any set of ω_{ij}s that satisfies the above constraints defines one valid tree.

Valid power configuration: The transmission power of each node N_i needs to be configured such that its transmitted packets can reach to its parent and all its child nodes. Thus, N_i's transmission power should be determined by the maximum of the distances to its parent and child nodes, i.e., $max_{1 \le j \le m}(\omega_{ij} + \omega_{ji}) \cdot d_{ij}$.

Thus, we have the following constraint:

$$RF(PW_i) \ge RF_{recv}^{min} + 10\gamma \log_{10}(\max_{1 \le j \le m}(\omega_{ij} + \omega_{ji}) \cdot d_{ij}) + C,\ \forall i \in \{1, 2, .., m\}, \quad (8)$$

where $RF(PW_i)$ denotes the emitted RF signal strength with the node's transmission power PW_i, and RF_{recv}^{min} is the minimum received RF strength that the signal must have to achieve a certain bit rate, γ is the path loss exponent whose value is normally in the range of 2 to 4, and C is a constant which accounts for system losses. This is the simplest form of the log-distance path loss model to ensure successful transmission from N_i to all its parent and child nodes.

Valid duty-cycle scheduling: The given solution of (BI_k, SD_k)s for all the clusters is valid only if a nonoverlapping SD schedule is possible. Fortunately, we can use Koubaa et al.'s algorithm [9] that checks if there exists a non-overlapping SD schedule with the set of (BI_k, SD_k)s for all the clusters. Let us just intuitively explain their algorithm using the same example they used in [9]. Fig. 4 shows how Koubaa et al.'s algorithm works for the example set of (BI_k, SD_k)s for six clusters, i.e.,

$$\{cluster_1(16, 4), cluster_2(8, 1), cluster_3(16, 2), cluster_4(32, 1), cluster_5(32, 4), cluster_6(16, 2)\}$$

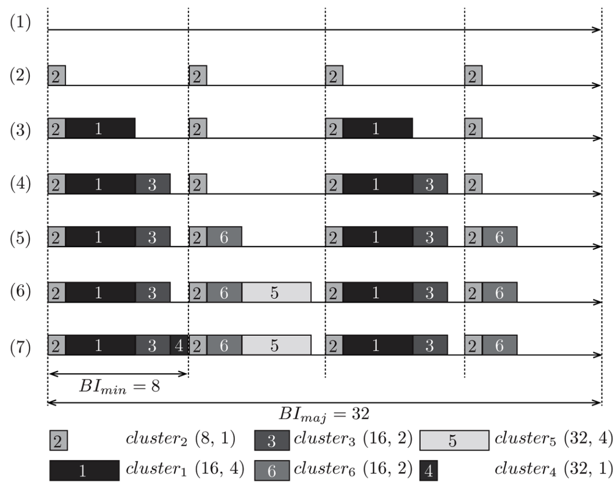

Fig. 4. Illustrative example of the Koubaa et al.'s algorithm.

where the smallest BI_k, called BI_{min}, is 8 and the largest BI_k, called BI_{maj}, is 32. The algorithm first sorts the given set of (BI_k, SD_k)s in ascending order of BI_k breaking ties for a large SD_k, resulting in the following sorted set:

$$\{cluster_2(8,\ 1),\ cluster_1(16,\ 4),\ cluster_3(16,\ 2),\ cluster_6(16,\ 2),\ cluster_5(32,\ 4),\ cluster_4(32,\ 1)\}.$$

With this sorted set of (BI_k, SD_k)s, the algorithm adds (BI_k, SD_k) to the schedule one by one. In Fig. 4, the algorithm starts with the empty schedule as shown in Line (1). On top of it, the algorithm takes the first element in the sorted set, i.e., $cluster_2(8,\ 1)$ and finds the first place where $SD_2 = 1$ can be placed at every $BI_2 = 8$ period for the entire duration of $BI_{maj} = 32$ without being overlapped with already placed SDs. Line (2) shows such placement for $cluster_2(8,\ 1)$. Succeeding lines of the figure show such placements for $cluster_1(16,\ 4)$, $cluster_3(16,\ 2)$, $cluster_6(16,\ 2)$, $cluster_5(32,\ 4)$, and $cluster_4(32,\ 1)$, in sequence. Finally, Line (7) results in the non-overlapping SD schedule found by Koubaa et al.'s algorithm for the given example set of (BI_k, SD_k)s. If the algorithm cannot find such a non-overlapping schedule, we simply say that the valid duty-cycle scheduling constraint cannot be met.

In addition, SD_k and GTS_k^j are related as depicted in Fig. 3 and the relation is formulated as follows:

$$SD_k \geq CAP_k + GTS_k^k + \sum_{j=1}^{m} \omega_{kj} GTS_k^j$$
$$= BI_k \cdot MIN_{nonRealTime} + GTS_k^k + \sum_{j=1}^{m} \omega_{kj} GTS_k^j . \quad (9)$$

Also, there is a restriction on validity of the GTS allocation by the IEEE 802.15.4 standard. Specifically, IEEE 802.15.4 specification limits the number of GTSs within an SD as 7. Thus, we have the following constraint:

$$\delta(GTS_i^i) + \sum_{j=1}^{m} \omega_{ij} \delta(GTS_i^j) \leq 7,\ \forall i \in \{1,\ 2,\ ..,\ m\}\ , \quad (10)$$

where $\delta(x)$ is one if $x > 0$ and zero otherwise.

End-to-end deadline guarantee: For the above valid solution of cluster-tree, (i.e., ω_{ij}s), power configuration, (i.e., PW_is), and duty-cycle scheduling, (i.e., BI_ks, SD_ks, and GTS_k^j s), we have to finally check if the solution can guarantee the end-to-end deadlines of all the real-time flows $\Gamma = \{f_1, f_2, \cdots, f_n\}$. For this check, let us explain how we can calculate the worst-case end-to-end delay for each flow f_a from the source node src_a to the destination node dst_a.

Thanks to the tree-based routing, once a cluster-tree topology is given, we can determine the routing path for each flow f_a. In the example of Fig. 2(b), the routing path of flow f_a from N_5 to N_2 is $N_5 \rightarrow N_4 \rightarrow N_3 \rightarrow N_2$. Using this routing path information, the worst case end-to-end delay of f_a can be computed by adding up the worst-case perhop

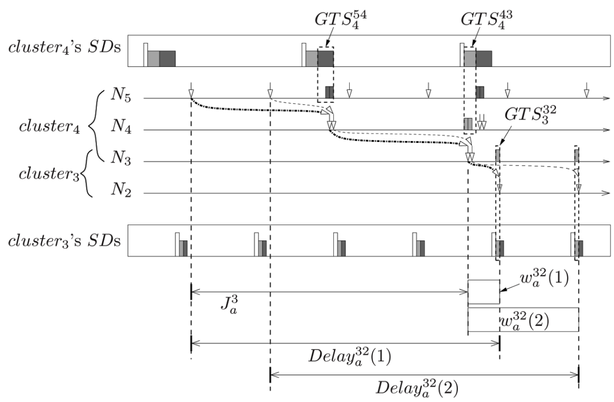

Fig. 5. Calculation of worst-case end-to-end delay. GTS: guaranteed time slot, SD: superframe duration.

delays as in Tindell et al.'s end-to-end response time analysis [10].

Let us explain this calculation for the example flow f_a in Fig. 2(b). The flow f_a's packet arrivals and transmissions at each hop from N_5 to N_2 are depicted in Fig. 5. At the source node N_5, packets of f_a are generated periodically as marked by the down-arrows in the N_5's timeline. However, each packet is transmitted using GTS_4^{54} as marked by a small shaded box in the N_5's time line. If there is more than one flow that passes through the link from N_5 to N_4, packets of multiple flows need to be properly scheduled by N_5. For packet scheduling, we use non-preemptive fixed priority scheduling assuming that packets from each flow have a pre-fixed priority. The pending packets are sorted by their priorities and the GTS is used to transmit the pending packets in the priority order. Due to the GTS waiting and the packet scheduling from multiple flows, the packet arrivals at the second hop are not periodic anymore as shown in the N_4's timeline. Therefore, more than one packets, generally q packet, from the same flow can be pending at the time of GTS. Such a situation of q packets from the same flow is well studied by Tindell's analysis at each hop to calculate the worst case end-to-end delay. Leveraging Tindell's approach, the worst-case end-to-end delay, denoted by WCD_a for flow f_a, can be computed by solving the following equations at each hop of the end-to-end path for f_a. The following equations are for the case of the hop from N_i to N_j where N_i is the cluster head. The other case where N_j is the cluster head can be similarly computed and we skip it.

$$w_a^{ij}(q) = \left\lceil \frac{q \cdot C_a + B_a^{ij} + \sum_{\forall b \in hp_a^{ij}} \left\lceil \frac{J_b^i + w_a^{ij}(q)}{Pb} \right\rceil \cdot C_b}{GTS_i^{ij}} \right\rceil \cdot BI_i . \quad (11)$$

$$WCD_a^{ij} = \max_{q=1,2,...} J_a^i + w_a^{ij}(q) - (q-1) \cdot p_a , \quad (12)$$

where

- hp_a^{ij} is the set of higher priority flows than f_a that pass through the hop from N_i to N_j. This set can easily be found thanks to the tree-based routing of all the flows.
- C_a is the transmission time of a packet of f_a. This is simply given as the packet size $size_a$ divided by the transmission rate R, that is, $C_a = size_a/R$.
- B_a^{ij} is the blocking time due to the non-preemptive transmission of a packet of a lower priority flow. It is given as the largest transmission time out of all the packets of lower priority flows that pass through the hop from N_i to N_j.
- J_a^i is the worst case arriving jitter of f_a due to scheduling delays at the hops before N_i. It is given as the worst case delay of f_a before it reaches N_i.

In Eq. (11), the intuitive meaning of $w_a^{ij}(q)$ is the length of the worst case time from the arrival of the first packet of f_a at N_i until completely transmitting first q packets of f_a to N_j. Fig. 5 shows examples of $w_a^{32}(1)$ and $w_a^{32}(2)$ at the hop from N_3 to N_2. In general, $w_a^{ij}(q)$ can be calculated by adding 1) the total time for transmitting q packets, 2) blocking time B_a^{ij} due to the already started transmission of a lower priority packet, and 3) largest possible delay due to higher priority packets that arrive before the complete transmission of q packets of f_a, which is given as $\sum_{\forall b \in hp_a^{ij}} \left\lceil \frac{J_b^i + w_a^{ij}(q)}{Pb} \right\rceil \cdot C_b$. In this term, $\left\lceil \frac{J_b^i + w_a^{ij}(q)}{Pb} \right\rceil$ is the largest possible number of packets of flow f_b during the time window $w_a^{ij}(q)$ assuming the worst-case scenario that the first packet of f_b is delayed the most, (i.e., J_b^i), and then the succeeding packets arrive with the maximum rate, (i.e., $1/p_b$) and, thus, the packet arrivals from f_b are most packed with the time window $w_a^{ij}(q)$. Note that $w_a^{ij}(q)$ appears in both sides making Eq. (11) recursive. This recursive equation can be solved iteratively starting with the initial assumption of $w_a^{ij}(q) = \left\lceil \frac{q \cdot C_a + B_a^{ij}}{GTS_i^{ij}} \right\rceil \cdot BI_i$ until $w_a^{ij}(q)$ does not further increase by additional packets of higher priority flows.

In Eq. (12), if we add the worst case delay of f_a until reaching N_i, i.e., J_a^i, to $w_a^{ij}(q)$, the result $J_a^i + w_a^{ij}(q)$ becomes the worst case time between the first packet generation time at the source node and the complete transmission time of the first q packets at the hop from N_i to N_j. Fig. 5 shows $J_a^3 + w_a^{32}(1)$ and $J_a^3 + w_a^{32}(2)$ for the example hop from N_3 to N_2. Thus, q-th packet's delay denoted by $Delay_a^{ij}(q)$ until reaching N_j is $J_a^i + w_a^{ij}(q) - (q-1) \cdot p_a$ as shown in Fig. 5. By picking the largest value of $J_a^i + w_a^{ij}(q) - (q-1) \cdot p_a$ out of all possible $q = 1, 2, \cdots$, the worst case delay WCD_a^{ij} until reaching N_j can be found using Eq. (12). Fortunately, Tindell and Clark [10] proved that such largest value exists out of $q = 1, 2, \cdots, Q$ where Q is the first integer that satisfies $w_a^{ij}(Q) \le Q \cdot p_a$. Therefore, it is sufficient to check only a finite number of q.

By repeatedly using the above equations from the source node to the destination node for each real-time flow f_a, we can compute the worst-case end-to-end delay WCD_a. If the computed end-to-end delay WCD_a is less than or equal to the end-to-end deadline D_a, we conclude that all the packets of f_a can be delivered to the final destination before their deadlines.

By repeating this check for all the real-time flows, we can verify whether or not the given solution of the cluster-tree, i.e., ω_{ij}s, power configuration, i.e., PW_is, and duty-cycle scheduling, i.e., BI_ks, SD_ks, and GTS_k^js, can deterministically guarantee the end-to-end deadlines of all the real-time flows.

C. Genetic Algorithm to Solve Our Optimization Problem

So far, we have formulated our problem as a formal optimization problem. However, it is a complex process to find a holistic solution for ω_{ij}s, PW_is, BI_ks, SD_ks, and GTS_k^js that maximizes the network lifetime, i.e., Eq. (3), while satisfying all the above constraints, that is, 1) valid tree construction constraints, 2) valid power configuration constraints, 3) valid duty-cycle scheduling constraints, and 4) end-to-end deadline guarantee constraints. Thus, it is computationally intractable to exhaustively search the entire solution space.

In order to manage such a high complexity of our holistic optimization problem, we use a genetic algorithm. For the genetic algorithm, we need a chromosome representation and a fitness function. Firstly, a chromosome string represents one possible solution of our holistic optimization problem. Thus, a chromosome string in our genetic algorithm is a complete set of all the parameters of cluster-tree construction, nodes' power configuration, and duty-cycle scheduling as shown in Fig. 6. Secondly, the fitness function is used to evaluate the quality of each solution, (i.e., the fitness of each chromosome string). As the fitness function, we use our objective function, i.e., the long-run average power function in Eq. (2).

With the chromosome representation and the fitness function, the genetic algorithm first forms the initial population of chromosome strings, which are seeded randomly in order to cover broad points of the entire solution space. Then, the genetic algorithm improves the chromosome strings in the initial population from generation to generation by repeating the following steps:

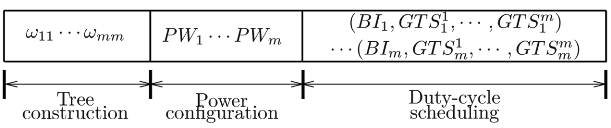

Fig. 6. Chromosome structure of the cluster-tree network. BI: beacon interval, GTS: guaranteed time slot.

• **Reproduction:** This step selects two chromosome strings as parents and combines them to create a new chromosome, which typically shares many characteristics of its parents. In our genetic algorithm, the standard weighted roulette wheel method [11] is used to select two chromosome strings with better fitness with higher probability.

• **Mutation:** This mutation step is taken probabilistically and once taken it gives a random perturbation to the reproduced chromosome string, which is necessary to avoid getting stuck at a local optima. The random perturbation is made by randomly choosing one value out of ω_{ij}s, PW_is, BI_ks, SD_ks, and GTS_k^js and then modifying it with a randomly generated value.

• **Repair:** The reproduced and mutated chromosome strings are mostly not feasible solutions for our problem. Therefore, the repair step is necessary after the reproduction and mutation steps. For an efficient repair of the unfeasible chromosome string, we employ a greedy repair method as follows:

- valid tree construction: If the ω_{ij}s in the chromosome string makes cycles in the resulting graph, we gradually remove edges from the longest one in the cycles until the graph becomes acyclic. If the ω_{ij}s result in a disconnected graph, we gradually add new edges from the pair of two closest nodes that do not make cycle until the graph becomes connected.
- valid power configuration: The transmission power PW_i of each node N_i is always repaired so that its signal can reach its farthest neighbor, based on the log-distance pass loss model in Eq. (8).
- valid duty-cycle scheduling: With the BI_ks, SD_ks in the chromosome string, if the duty-cycle scheduling is not feasible, we try increasing BI_ks or decreasing SD_ks so that the total sum of the duty-cycle, i.e., $\sum SD_k/BI_k$, can be reduced, which increases the chance of schedulability.
- end-to-end deadline guarantee: If the worst case end-to-end delay of any flow is greater than its end-to-end deadline, we decrease BI_ks along the flow's path while decreasing SD_ks and GTS_k^js accordingly in order to maintain the same ratio of SD_k/BI_k. This way, we can reduce the worst case end-to-end delay of the flow while keeping the same chance of duty-cycle schedulability.

• **Replacement:** After the new chromosome string is repaired as a feasible one, we replace a chromosome string in the population with the new one so that the population size can be kept the same. In order to select the chromosome string to be replaced, we employ an elitism strategy as in [11] because it ensures that the best chromosome string in the current generation always survives into the succeeding generation.

This repetitive evolution process of the population terminates either at a sufficient number of generations 50 in our case or when a specified percentage 70% in our case of the chromosome strings have the same best fitness.

V. EXTENSION TO SPATIAL REUSE OF AN RF RESOURCE

Our proposed holistic optimization so far assumes that only one cluster can be active at a time to avoid collisions of nodes in different clusters. That is why it tries to find a non-overlapping SD schedule of all the clusters. However, if two clusters are geographically far apart from each other, a node in one cluster does not cause any collision with a node in the other cluster even if two nodes transmit the data through the same RF channel at the same time. This is called a *spatial reuse of an RF resource*.

In order to take advantage of the spatial reuse of the RF, this section addresses an extended holistic optimization problem where more than one cluster can be active at the same time as long as they do not cause a collision. For this, in addition to the above three-dimensional parameters of 1) cluster-tree construction, 2) duty-cycle scheduling, and 3) nodes' power configuration, we handle one more dimension of spatial reuse by introducing the cluster color code parameter χ_k for each cluster $cluster_k$. The cluster color code is defined as follows: only if two clusters can be active at the same time without causing a collision, their color codes can be the same.

This color code assignment depends on how the clusters are formed, i.e., cluster-tree construction, and how much power each node is using, i.e., nodes' power configuration. Also, depending on the color code assignment, we can have different duty-cycle scheduling since the SDs of the same color clusters can be overlapped. As a result, all the four problems of 1) cluster-tree construction, 2) duty-cycle scheduling, 3) nodes' power configuration, and 4) spatial reuse of RF are inter-related. Therefore, we need an extended holistic solution that optimally solves all the four inter-related problems together.

For such an extended holistic optimization, we can still use the same objective function in Eq. (2) and the same constraints for valid tree construction, valid power configuration, and end-to-end deadline guarantee as in Section IV. On top of that optimization framework, we add the following *valid coloring* constraints that validly relate the new parameter χ_k to existing parameters ω_{ij} and PW_i.

Valid coloring: A coloring solution, that is, the set of color codes denoted by $(\chi_1, \chi_2, \cdots, \chi_l)$ assigned to the set of clusters ($cluster_1, cluster_2, \cdots, cluster_l$) is valid if the following constraints hold. For each pair of a head node N_{k_a} $(1 \leq k_a \leq l)$ and its child node N_j, that is, $\omega_{k_a j} = 1$, their signal-to-interference-plus-noise ratio (SINR) should be greater than a threshold $SINR_{th}$ [12] for their successful communications even with the interferences from all the other clusters with the same color codes, i.e., $cluster_{k_b}(1 \leq$

$k_b \le l,\ k_b \ne k_a,\ \chi_{k_b} = \chi_{k_a}$). The SINR from N_{k_a} to N_j considering interferences from all the other clusters with the same color codes can be computed as follows:

$$SINR(N_{k_a} \to N_j) = \frac{PW_{k_a} d_{k_a j}^{-\gamma}}{N + \sum_{\forall k_b \in K} \max_{\forall i \in cluster_{k_b}} (PW_i d_{ij}^{-\gamma})},$$

$$K = \{k_b | 1 \le k_b \le l,\ k_b \ne k_a,\ \chi k_b = \chi k_a\} \quad (13)$$

where $PW_{k_a} d_{k_a j}^{-\gamma}$ is the signal level at the receiver N_j, N is ambient noise, and $\sum_{\forall k_b \in K} \max_{\forall i \in cluster_{k_b}} (PW_i d_{ij}^{-\gamma})$ is a conservative estimation of the total interference from all the same colored clusters. Similarly, the SINR from N_j to N_{k_a} can be computed as follows:

$$SINR(N_j \to N_{k_a}) = \frac{PW_j d_{jk_a}^{-\gamma}}{N + \sum_{\forall k_b \in K} \max_{\forall i \in cluster_{k_b}} (PW_i d_{ik_a}^{-\gamma})},$$

$$K = \{k_b | 1 \le k_b \le l,\ k_b \ne k_a,\ \chi k_b = \chi k_a\} \quad (14)$$

where $PW_j d_{jk_a}^{-\gamma}$ is the signal level at the receiver N_{k_a}, N is ambient noise, and $\sum_{\forall k_b \in K} \max_{\forall i \in cluster_{k_b}} (PW_i d_{ik_a}^{-\gamma})$ is a conservative estimation of the total interference from all the same colored clusters.

Thus, we have the following constraints:

$$SINR(N_{k_a} \to N_j) > SINR_{th} \text{ and}$$
$$SINR(N_j \to N_{k_a}) > SINR_{th},$$
$$\text{for all } (k_a, j) \text{ with } \omega_{k_a j} = 1 \quad (15)$$

By checking these constraints for all the pairs of a cluster head and a cluster member, we can verify whether the given coloring solution ($\chi_1, \chi_2, \cdots, \chi_l$) is valid in the sense that all the clusters with the same color code can be active at the same time without collisions, that is, their *SD*s can be overlapped.

In addition to introducing these new valid coloring constraints, the previous valid duty-cycle scheduling in Section IV needs to be modified in a way of allowing overlapping SDs of the same color clusters in order to take advantage of spatial reuse of RF. For this, we modify Koubaa et al.'s algorithm explained in Section IV to allow SD overlapping whenever possible. Again, let us just intuitively explain our modified algorithm using the same example they used in [9]. Fig. 7 illustrates how our modified scheduling algorithm works for the example set of (BI_k, SD_k)s for six clusters, i.e.,

$$\{cluster_1(16, 4), cluster_2(8, 1), cluster_3(16, 2), cluster_4(32, 1), cluster_5(32, 4), cluster_6(16, 2)\}$$

where the smallest BI_k, called BI_{min}, is 8 and the largest BI_k, called BI_{maj}, is 32. For this example, let us assume that $cluster_1$ and $cluster_3$ have the same color code, that is, $\chi_1 = \chi_3$ and $cluster_4$ and $cluster_6$ have the same color code, that is, $\chi_4 = \chi_6$. The color codes of all the other clusters differ. For this given problem, as in Koubaa et al.'s algorithm, our modified algorithm also first sorts the given set of (BI_k, SD_k)s in ascending order of BI_k breaking ties for a large SD_k, resulting in the following sorted set:

$$\{cluster_2(8, 1), cluster_1(16, 4), cluster_3(16, 2), cluster_6(16, 2), cluster_5(32, 4), cluster_4(32, 1)\}.$$

With this sorted set of (BI_k, SD_k)s, our modified algorithm adds (BI_k, SD_k) to the schedule one by one just like Koubaa et al.'s algorithm. However, the difference is that our modified algorithm tries to place SD_k overlapping with already placed SDs if the color codes are the same. For example, when we add (BI_3, SD_3) in Line (4) of Fig. 7, our algorithm places SD_3 at every period BI_3 overlapping with already placed SD_1s because $\chi_3 = \chi_1$. Also, when we add (BI_4, SD_4) in Line (7), our algorithm places SD_4 at every BI_4 overlapping with SD_6 because $\chi_4 = \chi_6$. As a result, our algorithm can find the final *SD* schedule where *SD*s for the same color clusters possibly overlaps.

Since our modified algorithm allows SD to overlap whenever possible, a given set of (BI_k, SD_k)s is more likely to turn out valid by our algorithm even if it turns out invalid by Koubaa et al.'s algorithm. We can solve this extended holistic optimization formulation with the similar genetic algorithm as in Section IV. The only difference is the extension of the chromosome representation as in Fig. 8 including the color code parameters.

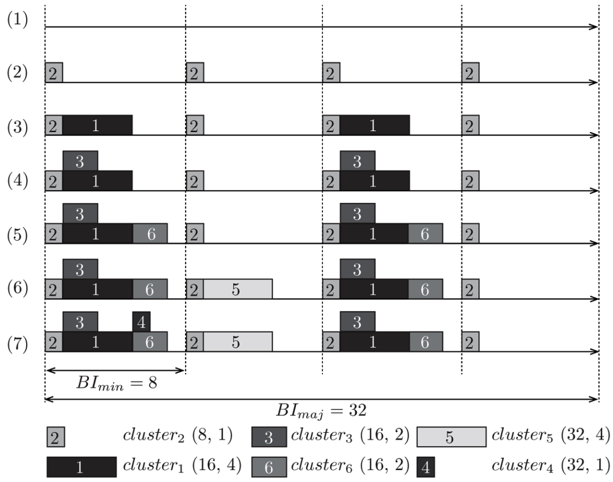

Fig. 7. Illustrative example of the spatial reuse scheduling algorithm. BI: beacon interval.

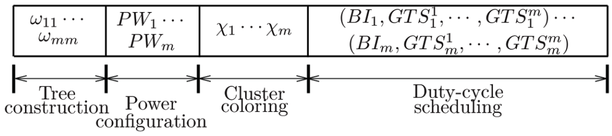

Fig. 8. Extended chromosome structure of the cluster-tree network. BI: beacon interval, GTS: guaranteed time slot.

Now, this chromosome string is a complete set of all parameters of cluster-tree construction, nodes' power configuration, duty-cycle scheduling, and cluster coloring. With this extended chromosome string, our genetic algorithm evolves the chromosome string to find the chromosome string with the best fitness for which we use the same fitness function in Eq. (2). In this evolution process, our genetic algorithm checks whether a chromosome string is a feasible solution or not by checking the original constraints of valid tree construction, valid power configuration, and end-to-end deadline guarantee in Section IV and the new and modified constraints of valid coloring and valid duty-cycle scheduling explained in this section.

VI. EXPERIMENTS

This section investigates how much improvement our holistic optimization can make. For this investigation, we consider 36 ZigBee nodes, i.e., $m = 36$, evenly placed in a 1200 m × 1200 m rectangular area. For each sensor node, we assume the TI CC2520 RF transceiver [13], for which the transmission rate R is 250 kbps, the packet reception power PW_{recv} is 55.5 mW, the range of controllable transmission power PW_i is (48.6 mW, 100.8 mW). Regarding the log-distance path loss model in Eq. (8), the CC2520 RF transceiver's minimum received RF strength RF^{min}_{recv} is -85 dBm. For the path loss exponent γ and the system loss constant C, we assume 3.5 and 0, respectively, as in [14].

On top of these sensor nodes, we randomly generate periodic real-time flows. More specifically, each flow f_a's packet size, $size_a$, is randomly generated following uniform distribution in the range of (7000 bits, 8000 bits). The period p_a and the end-to-end deadline D_a are also randomly generated following the uniform distribution in the ranges of (1000 ms, 2000 ms) and (3000 ms, 4000 ms), respectively. The flow f_a's source node src_a and destination node dst_a are also randomly picked out of the above 36 nodes. In the following, we consistently use these parameters for the random flow generations if not otherwise mentioned.

A. Experiments with No Spatial Reuse of an RF Resource

With the above settings, we compare the following three approaches assuming no spatial reuse of the RF resource:

• Han's approach [1] that only optimizes the duty-cycle scheduling assuming that the cluster-tree and nodes' powers are given. For the cluster-tree, we assume the minimum spanning tree construction in [15]. For the nodes' powers, we assume the maximum transmission power for every node.

• Our approach called 'optimal clustering and max power' that optimizes both duty-cycle scheduling and cluster-tree construction while using maximum power for all the nodes.

• Our approach called 'optimal clustering and optimal power' that optimizes duty-cycle scheduling, cluster-tree construction, and nodes' power configuration.

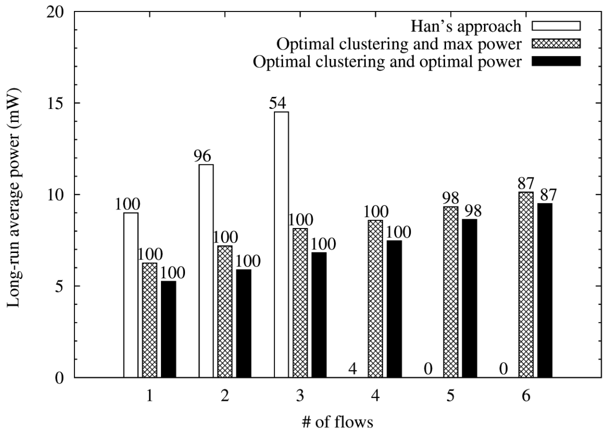

Fig. 9. Long-run average power of the bottleneck node as a function of the number of flows.

Fig. 9 compares the three approaches in terms of the long-run average power of the most bottleneck node, i.e., $max_{1 \leq i \leq m} avgPW(N_i)$ as defined in Eq. (2), as varying the number of real-time flows, i.e., n from 1 to 6. For the given number of real-time flows, i.e., n, we generate 100 problems with n real-time flows randomly generated as explained above. The number on each bar in Fig. 9 is the number of problems for which feasible solutions are found with the corresponding approach and each bar represents the average of the solutions found. As the number of flows increases, that is, as the overall system workload increases, the long-run average power also increases for all three approaches (For Han's approach, the bar drops when the number of flows is 4, i.e., $n = 4$. This is because Han's approach finds feasible solutions only for four problems out of 100 problems. The average of these four solutions does not well reflect the trend of the majority). However, the slopes of the increase are significantly different. For Han's approach, the power consumption of the most bottleneck node increases fast, since it does not distribute the flows by leveraging the freedom of tree clustering. For the same reason, Han's approach gets harder to find a feasible solution as the number of flows becomes larger. As a result, when $n > 4$, Han's approach cannot find feasible solutions for any of the 100 problems.

On the other hand, our approach 'optimal clustering and max power' can optimally leverage the freedom of tree clustering and, hence, distribute the flows well. Thus, the load given to the most bottleneck nodes can be limited even when the number of flows gets larger. Due to this reason, the gap between Han's approach and our

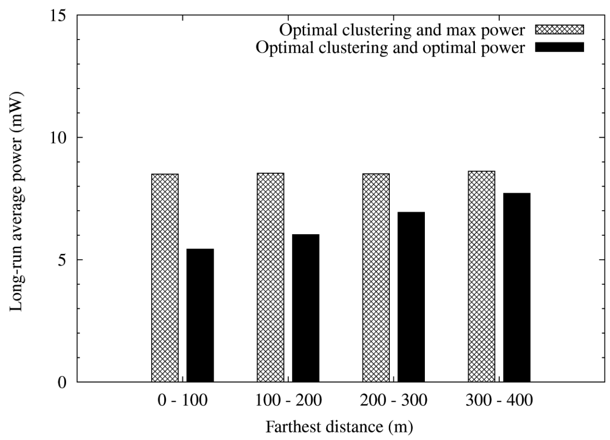

Fig. 10. Long-run average power of the bottleneck node as a function of the farthest distance.

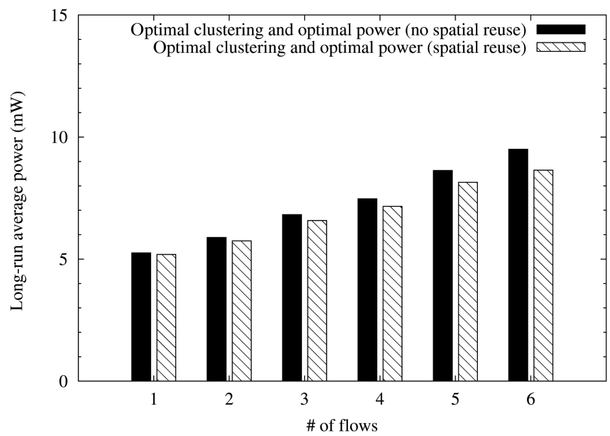

Fig. 11. Long-run average power of the bottleneck node as a function of the number of flows.

approach 'optimal clustering and max power' becomes larger as the number of flows increases. In addition, our approach 'optimal clustering and optimal power' can further save the power by using adequate power only rather than the max power.

The saving by our optimal power configuration largely depends on the farthest distance between a parent and its child among all parent-child pairs in the cluster-tree network. In other words, if the farthest distance is small, the parent and child can communicate with much less power than the max power. However, if the farthest distance is large, they need to use the max power anyway to reach each other. Hence the optimal power should be set to the same as the max power. In order to show this, Fig. 10 compares the two approaches for 500 random problems with a fixed number of flows, i.e., $n = 4$. In each bin of x-axis, i.e., (0-100), (100-200), (200-300), and (300-400), we put the solutions with the farthest distance in the corresponding range and average their fitness, i.e., long-run average power of the most bottleneck node. In the case of 'optimal clustering and max power', the long-run average power is almost constant regardless of the farthest distance. This is because all the nodes use the maximum transmission power without any power optimization. On the other hand, in the case of 'optimal clustering and optimal power', the long-run average power is much smaller when the farthest distance is small. This clearly shows that the optimal power configuration can make non-trivial power savings depending on the given problem settings of nodes and flows, especially when our solution can address the given problem with the farthest distance that is small.

B. Experiments with Spatial Reuse of an RF Resource

In this section, we investigate the improvement by the spatial reuse of the RF resource. For the experiments with the spatial reuse of the RF resource, the SINR interference model in Eq. (15) assumes the SINR threshold $SINR_{th}$ of 32 as in [12] and the ambient noise N of 0 as in [16]. With the spatial reuse of the RF resource, we can expect two consequences: further saving of the long-run average power and an increase of the effective capacity of the sensor network.

In order to investigate the first consequence, Fig. 11 compares the long-run average powers by our 'optimal clustering and optimal power' approach with and without the spatial reuse of the RF resource, as increasing the number of real-time flows, i.e., n from 1 to 6. Each bar in Fig. 11 is again the average for the 100 random problems with 36 nodes and n flows as mentioned above. As the number of flows increases, the long-run average power of the bottleneck node increases for both cases of spatial reuse and no spatial reuse. However, the gap between 'optimal clustering and optimal power (no spatial reuse)' and 'optimal clustering and optimal power (spatial reuse)' becomes larger as the number of flows increases. This can be explained as follows: with no spatial reuse, the number of clusters for which duty cycles are schedulable is quite limited. Thus, it should serve the given flows with a limited number of clusters forming large size clusters. Thus, the nodes in the large size clusters should use a large transmission power. This makes the fast increase of the long-run average power as the number of flows increases. On the other hand, with the spatial reuse, we can form a larger number of smaller size clusters thanks to the potential of overlapping SDs of multiple clusters. This makes the less severe increase of the long-run average power as shown in the Fig. 11.

Although the gap between the spatial reuse and the no spatial reuse does not on average look particular significant in Fig. 11. If we look at individual problems, we can observe significant gaps for many cases. For this, Fig. 12 dots the ratios of 'optimal clustering and optimal power (spatial reuse)' to 'optimal clustering and optimal power

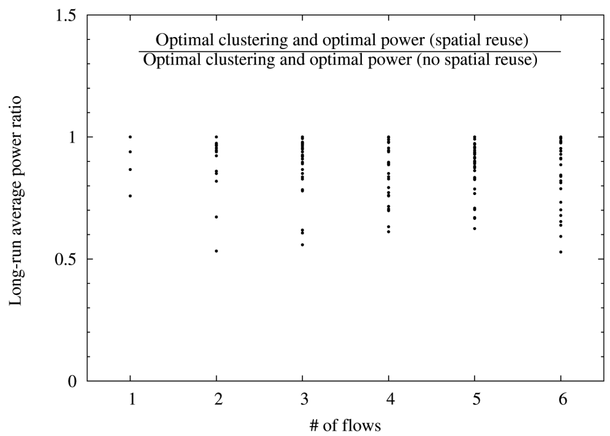

Fig. 12. Long-run average power ratio of 'optimal clustering and optimal power (spatial reuse)' to 'optimal clustering and optimal power (no spatial reuse)'.

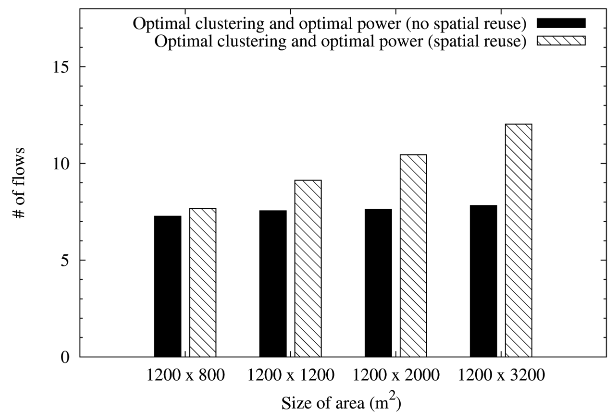

Fig. 13. Maximum affordable number of flows as a function of the size of area.

(no spatial reuse)' for the 100 individual problems, which are averaged in Fig. 11. In Fig. 12, we can see that spatial reuse can save more than 20% of the long-run average power of the no spatial reuse for many problems. For certain problems, the spatial reuse can save up to 50% of the long-run average power.

In order to investigate the second consequence, Fig. 13 compares the spatial reuse and the no spatial reuse in terms of the maximum affordable number of flows as varying the size of the sensor network area, i.e., 1200 m × 800 m, 1200 m × 1200 m, 1200 m × 2000 m, and 1200 m × 3200 m covered by the increasing number of evenly placed sensor nodes, i.e., m = 24, 36, 60, and 96. To find the maximum affordable number of flows, we prepare 30 randomly generated flows in advance for each size of area and add flows to the problem one-by-one in the same order until no more flows can be added to generate a feasible solution. We repeat this 100 times with 100 different sets of 30 prepared random flows. Each bar in Fig. 13 shows the average of the 100 experiments. For the no spatial reuse, the maximum affordable number of flows is almost constant regardless of the size of the area, since it does not take advantage of concurrent scheduling of faraway nodes with spatial reuse of the RF resource. On the other hand, for the spatial reuse, the maximum affordable number of flows increases as the size of the area increases since the spatially distributed flows in a larger area can be concurrently scheduled taking advantage of the spatial reuse of the RF resource.

VII. CONCLUSION

In this paper, we propose a holistic optimization method that builds the optimal IEEE 802.15.4/ZigBee network with on-time delivery of all real-time data while maximizing the lifetime of network. The holistic optimization method jointly addresses three related problems: logical cluster-tree construction, power configuration for nodes, and duty-cycle scheduling.

We first formulated the holistic optimization problem as a formal optimization problem. Thanks to this formal problem formulation, we could solve the holistic optimization problem with a genetic algorithm. Our experimental study shows that beyond the improvement through the optimal duty-cycle scheduling which has been performed in previous studies, the optimal tree-clustering and optimal nodes' power configuration can further significantly improve the lifetime of real-time the IEEE 802.15.4/ZigBee network. This justifies our holistic optimization framework that simultaneously optimizes duty-cycle scheduling, cluster-tree construction, and nodes' power configuration.

Currently, we are applying the proposed holistic optimization to build a ZigBee network for real-time monitoring of a large scale building. In the future, we plan to extend the optimization framework targeting the coexistence of real-time flows with deterministic and stochastic deadline guarantee requirements.

ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R0126-15-1105, Real-Time System SW on Multicore and GPGPU for Unmanned Vehicles). The ICT at Seoul National University provided research facilities for this study.

REFERENCES

1. J. Han, "Global optimization of ZigBee parameters for end-to-end deadline guarantee of real-time data," *IEEE Sensors*

Journal, vol. 9, no. 5, pp. 512-514, 2009.
2. S. C. Ergen and P. Varaiya, "Energy efficient routing with delay guarantee for sensor networks," *Wireless Networks*, vol. 13, no. 5, pp. 679-690, 2007.
3. M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," in *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS2002)*, Austin, TX, 2002, pp. 39-48.
4. T. Watteyne, I. Augè-Blum, and S. Ubèda, "Dual-mode real-time mac protocol for wireless sensor networks: a validation/simulation approach," in *Proceedings of the 1st International Conference on Integrated Internet Ad Hoc and Sensor Networks (InterSense2006)*, Nice, France, 2006.
5. O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in *Proceedings of the 14th IEEE International Workshop on Quality of Service (IWQoS2006)*, New Haven, CT, 2006, pp. 83-92.
6. T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor network," in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, Providence, RI, 2003, pp. 46-55.
7. E. Felemban, C. G. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Miami, FL, 2005, pp. 2646-2657.
8. ZigBee Alliance, "ZigBee specification," http://www.zigbee.org.
9. A. Koubaa, A. Cunha, and M. Alves, "A time division beacon scheduling mechanism for IEEE 802.15.4/ZigBee cluster-tree wireless sensor networks," in *Proceedings of the 19th Euromicro Conference on Real-Time Systems (ECRTS)*, Pisa, Italy, 2007, pp. 125-135.
10. K. Tindell and J. Clark, "Holistic schedulability analysis for distributed hard real-time systems," *Microprocessing & Microprogramming*, vol. 40, no. 2–3, pp. 117-134, 1994.
11. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
12. M. Hossian, A. Mahmood, and R. Jäntti, "Channel ranking algorithms for cognitive coexistence of IEEE 802.15.4," in *Proceedings of the 20th IEEE Personal, Indoor, Mobile Radio Communications (PIMRC)*, Tokyo, Japan, 2009, pp. 112-116.
13. Texas Instruments, "CC2520 datasheet 2.4 GHz IEEE 802.15.4/ZigBee RF transceiver," http://www.ti.com/lit/ds/symlink/cc2520.pdf.
14. V. S. Abhayawardhana, I. J. Wassell, D. Crosby, M. P. Sellars, and M. G. Brown, "Comparison of empirical propagation path loss models for fixed wireless access systems," in *Proceedings of the 61st IEEE Vehicular Technology Conference (VTC2005-Spring)*, Stockholm, Sweden, 2005, pp. 73-77.
15. R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, no. 6, pp. 1389-1401, 1957.
16. O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric SINR," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Montreal, Canada, 2007, pp. 100-109.

Kang-Wook Kim

Kang-Wook Kim received the B.S. degree in computer science and engineering in 2009 from Seoul National University, Korea, where he is currently working toward the Ph.D. degree in the School of Computer Science and Engineering. His current research interests include indoor localization, sensor networks, mobile platform, and real-time systems.

Myung-Gon Park

Myung-Gon Park received the B.S. degree in information and computer engineering in 2009 from Ajou University, Korea and M.S. degree in computer science and engineering in 2012 from Seoul National University, Korea. He is currently working at SAP Labs Korea. At SAP, he develops SAP S/4HANA database that performs parallel in memory processing of huge data sets to offer real-time responses for analytic queries.

Junghee Han

Junghee Han received the B.S. and M.S. degrees from Seoul National University and the Ph.D. degree from the University of Michigan. She is currently an associate professor at Korea Aerospace University. She was a senior engineer at Samsung Electronics and a research specialist of biomedical informatics at the Ohio State University. Her research interests include grid computing, network security, and wireless sensor networks. She is a member of the IEEE.

Chang-Gun Lee

Chang-Gun Lee received the B.S., M.S., and Ph.D. degrees in computer engineering from Seoul National University, Korea, in 1991, 1993, and 1998, respectively. He is currently a professor in the School of Computer Science and Engineering, Seoul National University. His current research interests include real-time embedded systems, cyber-physical systems, ubiquitous systems, QoS management, wireless ad hoc networks, and flash memory systems. He is a member of the IEEE and the IEEE Computer Society.