Maximizing Quality of Aggregation in WSNs Under Deadline and Interference Constraints

Hamed Yousefi[1], Masoud Mehrabi Koushki[2], Bahram Alinia[3], and Kang G. Shin[1]
[1]Department of Electrical Engineering and Computer Science, University of Michigan
[2]Department of Electrical and Computer Engineering, University of British Columbia
[3]Department RS2M, Institut Mines-Telecom, Telecom SudParis

***Abstract*—Maximizing quality of aggregation (QoA) is an essential requirement for real-time wireless sensor networks (WSNs) where the participation of *all* sensor nodes in data aggregation is hampered by the underlying sink deadline and interference constraints. This problem, however, remains unsolved under the physical interference model that captures the reality more accurately than the widely used graph-based models.**

In this paper, we formulate an optimization problem of maximizing QoA under deadline and interference constraints in commonly seen tree-based WSNs. We prove the problem to be NP-complete, and then propose a suboptimal scheduling algorithm which relies on a Markov approximation framework and modifies the matching graphs in order to handle the globally-imposed interference constraints. The problem and its solution are then coupled with successive interference cancellation (SIC) to improve QoA by increasing the number of concurrent transmissions. Our evaluation has shown the proposed solution to be effective under the physical interference model, both with and without SIC.

I. Introduction

Data aggregation has been used widely to save energy by reducing the number of packet transmissions in wireless sensor networks (WSNs). It is a key, but time-consuming function in tree-based WSNs, where packets are held for long enough at intermediate nodes to maximize aggregation efficiency at the expense of their delivery latency. However, real-time communications in WSNs are becoming important in emerging applications, such as target tracking, disaster and environmental monitoring, health monitoring, and battlefield surveillance. Obsolete information may be irrelevant and even harmful to the system monitoring and control, thus making *real-time data aggregation* essential.

Limiting the number of concurrent transmissions in the network to reduce interference is the main culprit for long *aggregation latency* at the data sink. Unfortunately, the interference problem is usually left to the MAC layer, incurring a significant amount of energy consumption and time latency for data aggregation. Hence, real-time data aggregation relies on the TDMA scheduling above the MAC layer to ensure interference-free transmissions [4], [13], [14], [19]. Most of the promising results thus far assume graph-based (i.e., hop- and range-based, and protocol) interference models), where the interference relationships can be represented by a conflict graph. However, this is an oversimplification of reality, where interference among concurrent transmissions are neither local nor pairwise, but global and additive [10], [13], [14], [21]. Thus, a signal is received successfully depending on the ratio of the received signal strength to the cumulative interferences caused by all concurrent transmissions plus ambient noise. This, known as the Signal-to-Interference-plus-Noise Ratio (SINR) model or the *physical interference* model, makes the previous approaches based on conflict graphs trivial or inapplicable.

In general, real-time data aggregation classifies the underlying optimization problems in two different categories according to the specific application needs.

- **Minimum Latency Aggregation Scheduling (MLAS):** While all sensor nodes need to report data periodically to the sink, the fundamental question is how fast information can be aggregated/gathered from a WSN rooted at the sink of aggregation [13], [14], [19]. However, some delay-sensitive applications may not even tolerate the latency obtained under MLAS. Therefore, a more important problem—as described next as the main focus of this paper—in the context of real-time data aggregation is to gather as much information as possible from the sensors within the maximum tolerable delay (a.k.a. deadline) of the application. Note that this problem is fundamentally different from MLAS in both the objective function and the constraint set, which calls for a new solution design.
- **Deadline-Constrained Aggregation Scheduling:** As an immediate consequence of real-time aggregation in delay-constrained WSNs, the imposed deadline prevents participation of all sensor nodes in interference-free data aggregation. It is therefore important to maximize *Quality of Aggregation* (QoA), defined as the number of nodes whose packets have been accounted for at the sink of aggregation [1], [8]. The fundamental question is then how to schedule transmissions (and which nodes to participate) under deadline and interference constraints. A few existing related approaches [2], [8], [9] tackled this problem only under the assumption of the simplest (i.e., one-hop[1]) interference model. To the best of our knowledge, there is no prior work on deadline-constrained aggregation scheduling problem under the SINR model, which is much harder due to the global and additive nature of the interference model. This is an important step forward, from the simplest to a very accurate interference model, for real-time data aggregation in WSNs.

[1]By the definition of the one-hop interference model, it only prevents the concurrent transmissions of children of the same parent in an aggregation tree.

This paper makes three main contributions:

(1) Maximizing QoA under the SINR model: We develop a framework that maximizes QoA under deadline and interference constraints. The problem is proved to be NP-complete, and a suboptimal solution for interference-free scheduling of packets in tree-based WSNs is proposed. We rely on a recently proposed Markov approximation framework [3] and modify the matching graphs to resolve the interferences under SINR while the participating nodes contribute to the optimal scheduling.

(2) Maximizing QoA under the SIC model: With the SINR model, a parent in the aggregation tree can successfully recover the signal from at most one child in each time slot. To overcome this limitation and improve the QoA by maximizing the number of concurrent transmissions, our problem is then coupled with successive interference cancellation (SIC), which is shown to be practical by experimental studies of WSNs [7]. It breaks the rule of one-time slot-one-sender barrier in each parent and lets it recover multiple individual signals received from multiple concurrently transmitting children, referred to as *meta-node*. In this new setting, dynamic formation and resolution of interfering group of concurrent meta-nodes are the major components of our solution.

(3) Theoretical analysis and experimental evaluation: We derive theoretical upper bounds on QoA under the SINR and SIC models. We also conduct several experiments to demonstrate the performance of our proposed algorithms. Applying globally-imposed interferences is shown to significantly reduce the optimal QoA given by [8] under the one-hop interference model. Moreover, by using multi-packet reception in our SIC algorithm, the proposed solution not only outperforms the SINR algorithm by an average of 44% but also exceeds the performance of [8] under various deadlines and network sizes.

The rest of this paper is organized as follows. Section II presents the system model and problem definition, and also studies the theoretical upper bounds on QoA. Our proposed solutions under SINR and SIC models are detailed in Section III and IV, respectively. Section V analyzes the complexities of the solutions. Our evaluation results are presented and analyzed in Section VI. The related work is summarized in Section VII. Section VIII discusses the limitations and extensions of our proposed approaches and finally, Section IX concludes the paper.

II. Problem Formulation

A. System Model

As in the state-of-the-art [2], [8], we model a WSN as a tree $G = (V \cup \{s\}, E)$, where s is the sink node and also the root of the tree. V is the set of N randomly distributed sensor nodes and E is the set of inter-communication links. A node may or may not be a source for a particular event. The system is time-slotted and the transmission of a packet takes exactly one time slot.

In a delay-constrained application, data needs to be aggregated at the sink before the imposed deadline of D time slots. However, an immediate consequence of deadline and interference constraints is that *all* sensor nodes cannot participate in the aggregation process. For each node $i \in V$, let $X[i, W_i]$ denote the maximum number of its source successors (i.e., all the source nodes in its subtree) that can be accounted for at node i if it is assigned a waiting of $W_i \in \{D-1, \ldots, 1, 0\}$ time slots for aggregation. Let $\mathcal{P}(i)$ and $\mathcal{C}(i)$ be the parent and the set of node i's children, respectively, and $V^{\text{leaf}} \subseteq V$ be the set of all leaf nodes. Moreover, $PATH(i) \subseteq V$ is the set of node i and all its predecessors toward the sink of aggregation. T_i is a binary decision variable such that $T_i = 1$ if node i is a source and $T_i = 0$ otherwise. We define $\vec{n} = [n_i, i \in V]$, where $n_i = 1$ if i is a participant (i.e., a source or relay node selected to participate in data aggregation). Let $V^{\text{src-par}} \subseteq V$ be the set of participating sources, formally defined by $V^{\text{src-par}} = \left\{ i \in V \mid T_i = 1 \text{ and } \prod_{j \in PATH(i)} n_j = 1 \right\}$, where i is a participating source if $T_i = 1$ and i as well as its predecessors participate in the aggregation.

B. Optimization Problem

1) Maximizing QoA under SINR: The cumulative interference from all concurrent transmissions is taken into consideration at each receiver. A transmission on link $(i,j) \in E$ is successful if the SINR at the receiver is above a certain threshold β [5], [6], [14]:

$$\frac{P_i / d_{ij}^{\alpha}}{N_0 + \sum_{k \in U_i} P_k / d_{kj}^{\alpha}} \geq \beta, \tag{1}$$

where U_i denotes the set of nodes transmitting concurrently with i, d_{ij} is the Euclidean distance between nodes i and j, and P_i is the transmission power of node i. Note that our problem does not involve power control, i.e., the nodes' transmission power is fixed and given as part of the input. Moreover, N_0 is the ambient noise power, α is the path loss ratio which typically ranges between 2 and 6, and the positive constant $\beta \geq 1$ is the SINR threshold for a successful transmission.

The objective is to maximize the number of participating sources (i.e., $|V^{\text{src-par}}|$) subject to deadline and interference constraints. This problem is formally expressed as:

$$Z_{\text{SINR}}: \ \text{maximize}_{\vec{n}, \vec{W}} \sum_{i \in V} T_i \prod_{j \in PATH(i)} n_j. \tag{2}$$

$$\text{s.t.} \quad W_i < W_j, \forall (i,j) \in E, j \in \{s\} \cup V \backslash V^{\text{leaf}}, \tag{3}$$

$$W_s = D \ , \ W_i \in \{D-1, \ldots, 1, 0\}, \quad \forall i \in V, \tag{4}$$

$$\vec{n} \in \{0,1\}^N, \tag{5}$$

$$\frac{P_i / d_{ij}^{\alpha}}{N_0 + \sum_{k \in U_i} P_k / d_{kj}^{\alpha}} \geq \beta . n_i, \quad \forall (i,j) \in E, \tag{6}$$

where decision variables are $\vec{n}$ and $\vec{W}$. Constraint (3) implies that in a feasible scheduling, no children can transmit before its parent; instead, a parent is required to wait for data from its participating children. It is straightforward to interpret constraints (4–6) based on our definitions. Constraint (6) also guarantees that no two children of the same parent can transmit concurrently because it causes interference, i.e., "at least" one transmission is not received successfully.

2) Maximizing QoA under SIC: When coupled with SIC, a receiver may recover multiple individual signals received from multiple concurrently transmitting children. Thus, SIC changes the definition of a successful transmission in which a parent can receive more than one packet transmission in the same time slot. The basic idea is to repeatedly decode the strongest signal received in a collision and remove it from the collided (mixed) signal [18]. To this end, an interference cancellation sequence needs to be identified, such that the following criterion will be met to cancel the i-th signal [7], [12]:

$$\frac{P_i/d_{ij}^{\alpha}}{N_0 + \sum_{k \in U_i - K_i} P_k/d_{kj}^{\alpha} + \sum_{l \in K_i, l \succ i} P_l/d_{lj}^{\alpha}} \geq \beta, \quad (7)$$

where $K_i = \{l \in \mathcal{C}(j) \mid W_l = W_i\}$ denotes the set of node j's children transmitting concurrently with node i (we call $i \cup K_i$ *concurrent children*), and $l \succ i$ means that transmission of i is canceled at j before that of l.

Finally, the problem under SIC can be formulated as:

$$Z_{\text{SIC}}: \ \text{maximize}_{\vec{n}, \vec{W}} \sum_{i \in V} T_i \prod_{j \in PATH(i)} n_j. \quad (8)$$

$$\text{s.t.} \quad \text{Constraints} \quad (3), (4), (5),$$

$$\frac{P_i/d_{ij}^{\alpha}}{N_0 + \sum_{k \in U_i - K_i} P_k/d_{kj}^{\alpha} + \sum_{l \in K_i, l \succ i} P_l/d_{lj}^{\alpha}} \geq \beta . n_i,$$

$$\forall (i,j) \in E, \quad (9)$$

where $\vec{n}$ and $\vec{W}$ are decision variables. Note that with SIC, an extra decoding delay is incurred to cancel a signal, but we will ignore this negligible effect ($\frac{3}{4096}$ time units in ZigBee, which is a common physical layer standard for WSNs [7]) in our analysis and evaluation.

C. QoA Upper Bound

Theorem 1: For an imposed deadline D, the QoA under SIC is upper bounded by $(M+1)^D - 1$, where M is the maximum number of children concurrently transmitting to the same parent.[2]

Proof: Considering the deadline and interference constraints, the maximum QoA is attained if all nodes are sources and scheduling packet transmissions under one-hop interference model does not cause any global interference. Thus, in the best case, at most M concurrent transmissions can be scheduled in every group of $M+1$ nodes (M children and their parent). In a network with n participating nodes, each of which needs to transmit exactly once, at most $n\frac{M}{M+1}$ nodes take the first slot, while $\frac{n}{M+1}$ nodes remain. This process continues in slot t ($1 \leq t \leq D$), where at most $n\frac{M}{M+1}(\frac{1}{M+1})^{t-1}$ nodes are scheduled for transmission and $n(\frac{1}{M+1})^t$ nodes remain. Finally, after deadline D, only the sink node remains. We thus have $n(\frac{1}{M+1})^D \leq 1$, which gives $n \leq (M+1)^D$. By removing the sink from the set of participants, the maximum QoA is bounded by $(M+1)^D - 1$, proving the theorem. ∎

SINR is a special case of SIC, where no two children of the same parent can concurrently transmit [12]. Thus, by setting M to 1 in the SIC bound, the maximum QoA under the SINR model is equal to 2^D-1. This bound has already been proved in [2] under the one-hop interference model, thus verifying our general QoA bound under SIC.

Example: Fig. 1 gives an example of a data aggregation tree where the sink deadline is $D = 2$ and all nodes are source. We want to demonstrate the impact of scheduling on QoA. Fig. 1(a) shows a possible assignment of waits ($W_2 = 1$ and $W_3 = 0$) which cannot maximize the number of participating sources in the aggregation. Instead, Fig. 1(b) shows one of the optimal choices which attains the highest QoA with 3 participating sources (indicated by gray nodes), where nodes 2 and 5 send their packets to separate parents in the first time slot concurrently (i.e., $W_2 = W_5 = 0$); then, in the second time slot, node 3 aggregates its own packet with that of node 5 and sends the result to the sink.

Fig. 1(c) shows the optimal choice under SIC with $M = 2$. For each parent under SIC, its children are partitioned into the groups of two concurrent children (referred to as *meta-nodes* and indicated by dashed ovals in the figure). Finally, under this optimal scheduling, 8 nodes participate in aggregation, which shows how using SIC can significantly improve QoA.

Next, we present our solutions for two problems Z_{SINR} and Z_{SIC} in Sections III and IV, respectively.

III. SINR Solution

The problem is first proved to be NP-complete, and then a heuristic solution is proposed to provide suboptimal interference-free scheduling.

Theorem 2: Z_{SINR} is NP-complete.

Proof: We first show that Z_{SINR} is in NP. Given an aggregation tree with N nodes and assigned waiting times from the set $\{0, 1, \ldots, D-1\}$, one can use the SINR Eq. (1) to verify whether or not the nodes with equal waiting can transmit concurrently. As there are less than N transmissions in each time slot, the verification costs $O(N^2 D)$.

The minimum latency aggregation scheduling for arbitrary aggregation trees is studied in [17]. This problem (a.k.a. MLAT) is a special case of MLAS, where the tree topology is given *a priori* instead of first constructing a tree in MLAS. A polynomial time algorithm is developed in [17] that reduces the *Partition* problem—which is known to be NP-hard—to MLAT (see [17] for a detailed proof of NP-hardness of MLAT). In the rest of our proof, we show that MLAT can be reduced to Z_{SINR} in polynomial time, proving that the reduction from the Partition problem to Z_{SINR} can also be done in polynomial time.

Consider MLAT for a given aggregation tree G_{MLAT} with N sensor nodes. Clearly, an upper bound of the minimum latency in MLAT is N. To reduce MLAT to Z_{SINR}, we construct the same tree in MLAT (with the same position of nodes)

[2]According to [12], $M \leq \lfloor \log_{1+\beta} \frac{P_M d^{-\alpha}}{N_0 \beta} + 1 \rfloor$, where P_M is the maximum transmission power and d is the minimum length of the links in the aggregation tree.

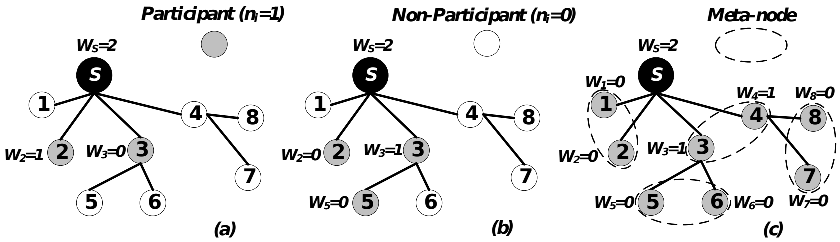


Fig. 1: Scheduling with $D = 2$ in an example aggregation tree: (a) a possible scheduling, $QoA = 2$; (b) QoA upper bound under SINR, $QoA = 3$; and (c) QoA upper bound under SIC with $M = 2$, $QoA = 8$

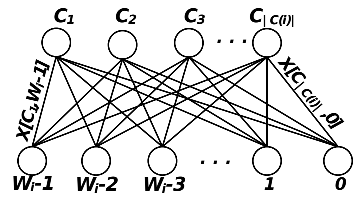


Fig. 2: Matching Graph

named G_{Z} and set all nodes as source. Let $\mathcal{A}$ be an optimal scheduling for Z_{SINR}, i.e., given an arbitrary tree, $\mathcal{A}$ maximizes the number of participating nodes in data aggregation (i.e., QoA) under the sink deadline D and the SINR model. We run algorithm $\mathcal{A}$ on G_Z for different deadline values starting from 1 to N until finding the least value of deadline (say D_{min}) that results in the participation of "all" nodes in the aggregation. Clearly, D_{min} is the answer (i.e., minimum latency) for MLAT in G_{MLAT}, and the scheduling returned by $\mathcal{A}$ in G_{Z} is also the optimal scheduling for MLAT in G_{MLAT} (note that finding D_{min} by running $\mathcal{A}$ can be done in $O(\log N)$ using binary search). Therefore, $\mathcal{A}$ solved MLAT (and hence, the Partition problem) in polynomial time which is a contradiction unless $P = NP$, thus completing the proof. ■

Note that although a solution for Z_{SINR} can also solve the MLAT problem under SINR, the reverse is not true. In other words, the solutions for MLAT—and generally MLAS—cannot be employed to solve Z_{SINR}.

General Idea: Our strategy is to derive the solution of *BASIC* [8]—which is under the one-hop interference model and gives an upper bound to our problem—and then resolve the globally-imposed interferences under the SINR model iteratively, on a slot-by-slot basis. In each iteration (corresponding to a specific time slot), the resolution process yields a new configuration of participating nodes and their schedules which are fed as inputs to the next iteration. This implies a complex *dependent* scheduling, as described later, for maximizing QoA in delay-constrained WSNs. Algorithm 1 shows an abstract view of our solution running in two steps: (1) *BASIC*, and (2) *Resolution* taking the global interference into account.

Algorithm 1: Our Algorithm

```
Input: Tree G, deadline D
Output: n⃗, W⃗
//BASIC
(n⃗, W⃗) = BASIC (G, D)
//Resolution
for i = D − 1, ..., 0 do
    (n⃗, W⃗) = Resolution (G, i, n⃗, W⃗)
```

A. Solution Details

Before detailing the algorithm, we describe how to construct a matching graph which is a core part of the solution in both *BASIC* and *Resolution* steps.

Maximum Weighted Matching (MWM): The problem of finding $X[i, W_i]$ is a maximum weighted matching problem in a bipartite graph, called *matching graph*, with two disjoint sets A and B as shown in Fig. 2. Here, $A = \mathcal{C}(i) = \{c_1, \ldots, c_{|\mathcal{C}(i)|}\}$ is the set of children of node i and $B = \{W_i - 1, W_i - 2, \ldots, 0\}$ represents their possible waiting times. The edge connecting a child node $a \in A$ to a waiting time $b \in B$ has a weight $X[a, b]$—which is the QoA provided by node a under deadline b. The set of edges and their weights in the bipartite graph are represented by $\mathcal{M} = \{(a, b) : a \in A,\ b \in B\}$ and X, respectively. In each parent, MWM assigns the waiting times to its children nodes such that 1) no two children have the same waiting time, 2) each child can be allotted at most one waiting time, and 3) the sum of weights on all selected edges is maximized. More formally, MWM can be represented as:

$$\mathsf{MWM}(A, B, \mathcal{M}, X): \quad \max_{\text{matching}\,\mathcal{O} \subset \mathcal{M}} \sum_{(a,b) \in \mathcal{O}} X[a, b].$$

Algorithm 2: BASIC

```
Input: Tree G, deadline D
Output: n⃗, W⃗
//Bottom-Up Procedure (leaves → root)
for all l ∈ V^leaf and W_l = 0, ..., D − 1 do
    X[l, W_l] = T_l
for all i ∈ V\V^leaf do
    for W_i = 0, ..., D − 1 do
        Find X[i, W_i] by solving
        MWM(C(i), {W_i − 1, W_i − 2, ..., 0}, M, X)
Find X[s, D] by solving MWM(C(s), {D − 1, D − 2, ..., 0}, M, X)
//Top-Down Procedure (root → leaves)
for all i ∈ {s} ∪ V\V^leaf do
    Use the solution of MWM(C(i), {W_i − 1, W_i − 2, ..., 0}, M, X)
    to select the participating children and assign them waiting times
```

1) ***BASIC:*** *BASIC* runs in two phases as shown in Algorithm 2. The first is a bottom-up procedure that calculates $X[i, W_i]$ for each node i and waiting time $W_i \in \{D-1, \ldots, 1, 0\}$. Unlike the original *BASIC* [8] in which $B = \{W_i - 1, W_i - 2, \ldots, W_i - \min(W_i, |\mathcal{C}(i)|)\}$, we also include $\{W_i - |\mathcal{C}(i)| + 1, \ldots, 1, 0\}$ in the matching graph even if $|\mathcal{C}(i)| < W_i$. This is because the related information is necessary in our resolution approach which modifies the matching graphs and replaces some wait assignments to improve QoA (see Section III-A2). After $X[i, W_i]$ is recursively calculated by solving MWM, the second phase is launched from the sink towards the leaves. In this top-down procedure, each parent i uses the MWM solutions to assign its children the final optimal waiting times under one-hop interference model such that the maximum number of sources in their corresponding subtrees can participate in data aggregation.

*2) **Resolution**:* The *BASIC* algorithm outputs a sequence of transmitter sets $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{D-1}$ such that data packets are concurrently transmitted from the nodes in $\mathcal{S}_i$ after waiting i time slots. Similarly, each solution $\mathcal{S}$ in our problem can be represented as $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{D-1}$. However, the initial $\mathcal{S}_i$ may not be a feasible solution under SINR, where globally-imposed interferences must be taken into consideration. Thus, as mentioned earlier, we need to develop an interference resolution algorithm to modify currently assigned waitings and make interference-free schedules while maximizing QoA.

We employ a top-down approach with D iterations (implemented by a "for" loop in Algorithm 1) starting from $D-1$ to 0. At each iteration i, we construct an interference-free set of concurrent transmitters $\widehat{\mathcal{S}}_i$ from $\mathcal{S}_i$. Therefore, after D iterations, all participating nodes are identified, resulting in a new set $\widehat{\mathcal{S}} = \widehat{\mathcal{S}}_0 \cup \widehat{\mathcal{S}}_1 \cup \cdots \cup \widehat{\mathcal{S}}_{D-1}$. It is important to note that (i) final set $\widehat{\mathcal{S}}_i$ may have no relation to $\mathcal{S}_i$, so $\widehat{\mathcal{S}}_i \subseteq \mathcal{S}_i$ does not hold in general, (the process to construct $\widehat{\mathcal{S}}_i$ will be described later in this section), and (ii) resolution of $\mathcal{S}_i$ can affect $\mathcal{S}_j,\ \forall j \leq i$, so $\vec{n}$ and $\vec{W}$ are updated after resolution in each time slot. With this dependent scheduling, almost from the initial rounds, we face entirely different input sets of transmitters from *BASIC* initial output sets for resolution in all time slots.

To achieve a near-optimal solution for this complex problem, we use the log-sum-exp approximation method. What follows shows how we apply a recent Markov approximation framework [3] and modify matching graphs to resolve the interferences under SINR in each time slot while the participating nodes contribute towards the optimal scheduling.

Markov Approximation: Markov approximation [3] is a general technique to approximately solve a wide range of combinatorial optimization problems. Its idea is to consider the solution space of the combinatorial problem as state space of the Markov chain. Then, finding the approximate solution requires random walks on the Markov chain while assuring a state with better result (higher QoA in our case) has higher chance to be visited.

Consider a scheduling problem of Z_{SINR} with a set of *feasible* solutions $\mathcal{F}$. Each solution $\mathcal{S} \in \mathcal{F}$ indicates the set of participants and their waiting times (i.e., the vectors $\vec{n}$ and $\vec{W}$). Let $\Pi_{\mathcal{S}}$ denote the obtained QoA when the system relies on scheduling $\mathcal{S}$. Then, the problem of maximizing QoA has the same optimal value of the following problem:

$$Z^{eq}_{\text{SINR}}: \quad \max_{p_{\mathcal{S}} \geq 0, \mathcal{S} \in \mathcal{F}} \sum_{\mathcal{S} \in \mathcal{F}} p_{\mathcal{S}} \Pi_{\mathcal{S}}, \qquad s.t. \quad \sum_{\mathcal{S} \in \mathcal{F}} p_{\mathcal{S}} = 1 \ ,$$

where $p_{\mathcal{S}}$ denotes the percentage of time that the system relies on solution $\mathcal{S}$. This maximization can be approximated by the log-sum-exp function [3] with a positive constant coefficient β_m (which controls the approximation accuracy) as:

$$Z^{\beta_m}_{\text{SINR}}: \max_{p_{\mathcal{S}} \geq 0, \mathcal{S} \in \mathcal{F}} \sum_{\mathcal{S} \in \mathcal{F}} p_{\mathcal{S}} \Pi_{\mathcal{S}} - \frac{1}{\beta_m} \sum_{\mathcal{S} \in \mathcal{F}} p_{\mathcal{S}} \log p_{\mathcal{S}},$$
$$s.t. \qquad \sum_{\mathcal{S} \in \mathcal{F}} p_{\mathcal{S}} = 1 \ .$$

Thus, we implicitly solve an approximated version of the main problem, off by an entropy term $-\frac{1}{\beta_m}\sum_{\mathcal{S}\in\mathcal{F}} p_{\mathcal{S}} \log p_{\mathcal{S}}$. The optimal value for $p_{\mathcal{S}}$ can be obtained by solving KKT conditions as:

$$p^*_{\mathcal{S}} = \frac{\exp\left(\beta_m \Pi_{\mathcal{S}}\right)}{\sum_{\widehat{\mathcal{S}} \in \mathcal{F}} \exp\left(\beta_m \Pi_{\widehat{\mathcal{S}}}\right)}, \quad \mathcal{S} \in \mathcal{F} \ . \tag{10}$$

We remark that the optimality gap is bounded by $\frac{1}{\beta_m} \log |\mathcal{F}|$ (we refer to [3] for details). The gap is clearly lower for a higher value of β_m, implying a more accurate approximation. It converges to zero as $\beta_m \to \infty$.

Markov Chain Design: The key step to leverage the Markov approximation is to design an application-specific Markov chain with state space being $\mathcal{F}$ and stationary distribution $p^*_{\mathcal{S}}$. This way, we find an exact solution for $Z^{\beta_m}_{\text{SINR}}$ which yields an approximate solution for the main problem $Z^{\text{eq}}_{\text{SINR}}$.

Each state $\mathcal{S}$ of the Markov chain is in the form of $\mathcal{S} = \mathcal{S}_0 \cup \cdots \cup \mathcal{S}_{D-1}$. Running the algorithm at time slot i, a next potential state is $\widehat{\mathcal{S}} = \mathcal{S}_0 \cup \cdots \cup \widehat{\mathcal{S}}_i \cup \cdots \cup \mathcal{S}_{D-1}$, where $\widehat{\mathcal{S}}_i$ is a permutation of $\mathcal{S}_i$. In other words, for a set of nodes concurrently transmitting after waiting i time slots, each Markov state S corresponds to a possible permutation of the nodes. Each permutation provides us with an order of resolving interferences among concurrent transmitters, so it can change the set of participating nodes in the aggregation and result in different QoA. Therefore, walking on different permutations is equal to hopping over different states. Now, by appropriately setting the transition rates among different states, the stationary distribution in (10) can be achieved. By the theoretical framework, and as in [1], [20], we set the transition rate from $\mathcal{S}$ to $\widehat{\mathcal{S}}$ as:

$$q_{\mathcal{S},\widehat{\mathcal{S}}} = \frac{1}{\exp\left(\alpha_m\right)} \frac{\exp\left(\beta_m \Pi_{\widehat{\mathcal{S}}}\right)}{\exp\left(\beta_m \Pi_{\mathcal{S}}\right) + \exp\left(\beta_m \Pi_{\widehat{\mathcal{S}}}\right)}, \tag{11}$$

where α_m is a positive constant and $q_{\widehat{\mathcal{S}},\mathcal{S}}$ is defined symmetrically. Finally, we need to ensure that the designed Markov chain has the stationary distribution in (10). According to [20], the following two conditions are sufficient to achieve this goal: (a) the state space should be connected and each state should be reachable (with a sequence of hops) from any other state, and (b) the detailed balance equation, $p^\star_{\mathcal{S}} q_{\mathcal{S},\widehat{\mathcal{S}}} = p^\star_{\widehat{\mathcal{S}}} q_{\widehat{\mathcal{S}},\mathcal{S}}$, is satisfied. In our designed Markov chain, any possible permutation of nodes is considered to build the next state of the Markov chain with a non-zero transition probability. Moreover, it can be verified that using (11), the balance equation is satisfied. We refer readers to [3] for detailed information on the Markov approximation.

Resolution Algorithm: Our interference resolution proceeds in two sequential phases for each time slot i as summarized in Algorithm 3:

(1) The first phase is a random walk on the Markov chain starting from an initial state towards those resulting better QoA. Each time, we temporarily move to a new state $\widehat{\mathcal{S}}$ and estimate how this change affects QoA in our aggregation scheduling (Alg. 3, Lines 4 and 5, respectively). We keep new state with a probability of $q_{\mathcal{S},\widehat{\mathcal{S}}}$ and switch back the previous

Algorithm 3: Resolution in each time slot i

Input: Tree G, Slot i, $\vec{n}$, $\vec{W}$, Constant T
Output: $\vec{n}, \vec{W}$

//Phase 1: Random Walk on Markov Chain
1 $\mathcal{S}_i \leftarrow \{k \,\epsilon\, V \mid W_k = i\}$
2 $t = 1$
3 **repeat**
4   $\widehat{\mathcal{S}}_i \leftarrow perm(\mathcal{S}_i)$ //A Possible Permutation
5   $q_{\mathcal{S},\widehat{\mathcal{S}}} = \frac{1}{exp(\alpha_m)} \frac{exp(\beta_m \Pi_{\widehat{\mathcal{S}}}(.))}{exp(\beta_m \Pi_{\mathcal{S}}(.)) + exp(\beta_m \Pi_{\widehat{\mathcal{S}}}(.))}$ //Using Alg.4
6   $r = rand(0, 1)$
7   **if** $r < q_{\mathcal{S},\widehat{\mathcal{S}}}$ **then**
8     $\mathcal{S} \leftarrow \widehat{\mathcal{S}}$
9   $t = t + 1$
until $t \leq T$

//Phase 2: Update (Participants and Waitings)
10 **for** *all* $j \in \mathcal{S}_i = \{i_1, i_2, \dots, i_{|\mathcal{S}_i|}\}$ **do**
11   **if** *matching graph of* $\mathcal{P}(j)$ *is modified in Alg. 4* **then**
12   Run *BASIC* on $\mathcal{P}(j)$'s subtree to update its participating successors and assign their waiting times

Algorithm 4: QoA estimation in each state $\mathcal{S}$ and matching graph modification in $\mathcal{S}_i$

Input: Tree G, Slot i, State $\mathcal{S}$ (including Set $\mathcal{S}_i$), $\vec{n}$, $\vec{W}$
Output: Estimated QoA Q
1 $R = \{\}$
2 $Q = X[s, D]$
3 **for** *all* $j \in \mathcal{S}_i = \{i_1, i_2, \dots, i_{|\mathcal{S}_i|}\}$ **do**
4   $R = R \cup \{j\}$
5   **while** R *is a case of interference* **do**
6   Remove link (j, i) from set $\mathcal{M}$ of links in matching graph of parent $\mathcal{P}(j)$
7   Solve MWM$(\mathcal{C}(\mathcal{P}(j)), \{W_{\mathcal{P}(j)} - 1, \dots, 1, 0\}, \mathcal{M}, X)$ and update Q according to the new wait assignments
8     $R = R \setminus \{j\}$
9   **if** *another child* $k \in \mathcal{C}(\mathcal{P}(j))$ *takes slot* i **then**
10       $R = R \cup \{k\}$
11       $j = k$

state $\mathcal{S}$ with a probability of $1 - q_{\mathcal{S},\widehat{\mathcal{S}}}$ (Alg. 3, Lines 6–8). Calculating the transition rates between two states requires obtaining their providing QoA while assuring an interference-free schedule. This is achieved by modifying the matching graphs of parents of interfering transmitters and estimating the resulting QoA after modification as described below and also shown in Algorithm 4.

Let i be the current slot in Algorithm 3. In each Markov state, we start with an empty set and add the transmitters one by one according to their priorities (i.e., underlying sequence of nodes in the permutation) subject to SINR. We also try to replace an interfering transmitter with a non-interfering child of the same parent. To achieve this goal, we remove link (j, i) from set $\mathcal{M}$ of links in matching graph of parent $\mathcal{P}(j)$ if node j's transmission causes interference (Alg. 4, Line 6). Based on the modified matching graph, which no longer supports perfect matching shown in Fig. 2, we run MWM for $\mathcal{P}(j)$ and see if another child k can take waiting i for transmission (Alg. 4, Lines 7–11). We again remove link (k, i) in case of interference, and continue this process (referred to as *replacement search*) until one (or no) child can successfully take slot i (this child, if any, will be added to the new set of transmitters (Alg. 4, Line 10)). It is worth noting that running MWM on the modified matching graphs in slot i may only affect the assignment of lower slots, i.e., slot l, $\forall l \leq i$ (not the higher ones already assigned to the nodes providing higher QoAs). This ensures a loop-free solution. Algorithm 4 ends after finding the maximum number of possible concurrent transmissions according to the nodes priorities. The QoA associated with each state is estimated based on new local wait assignments and the way they affect the number of participating source successors. The following example clarifies how we resolve interference in a state while modifying a matching graph.

Fig. 3(a) shows part of an example aggregation tree including concurrent transmitters after waiting 2 time slots under one-hop interference model. Considering the global interference under SINR, this scheduling is no longer a feasible solution to the problem. To resolve interferences, let $\{b_1, a_1, c_3\}$ be the current permutation in the underlying Markov chain. This permutation calls for consideration of a sequence of nodes, according to which child c_3 cannot transmit along with $\{b_1, a_1\}$. Using only the Markov framework yields a set $\mathcal{S}_2 = \{b_1, a_1\}$, where no more nodes take waiting 2 time slots. However, we try not only to find the best permutation, but also to fully utilize each waiting time by running a replacement search in the parents of the interfering transmitters, thus maximizing QoA. Then, to check for best possible replacement of c_3, we simply remove link $(2, c_3)$ from the matching graph of parent C as shown in Fig. 3(b), where running MWM in the new setting assigns $W = 2$ to c_1 as a feasible solution for the problem. Finally, as shown in Fig. 3(c), $\mathcal{S}_2 = \{b_1, a_1, c_1\}$ forms the new set of concurrent transmitters. We can then estimate QoA according to the new local updates (exchanging waiting in c_3 with c_1) in the current state in the underlying Markov chain.

(2) The second phase (Alg. 3, Lines 10–12) is launched after T rounds, in which the final permutation (say $\mathcal{S}_i$) is determined and the matching graphs are accordingly modified to remove interferences and possibly add the replacements. Here, each parent $\mathcal{P}(j) : \forall\, j \in \mathcal{S}_i$ with a modified matching graph needs to run the *BASIC* algorithm on its subtree to update the waiting times of its participating successors. Note that this procedure in slot i does not affect schedules in higher time slots $l > i$, which ensures a loop-free resolution.

Theorem 3: Algorithm 1 provides an interference-free scheduling for Z_{SINR}.

Proof: Consider a scheduling problem with the sink deadline D. The output of Algorithm 1 is a sequence of sets $\mathcal{S}_0, \mathcal{S}_2, \dots, \mathcal{S}_{D-1}$, where $\mathcal{S}_i$ identifies the set of nodes concurrently transmitting after waiting i. Due to transmissions in different time slots, no transmitter in $\mathcal{S}_i$ interferes with one in $\mathcal{S}_j$, $i \neq j$. Therefore, it is sufficient to show that transmissions associated with a set $\mathcal{S}_i$ are interfere-free. In our solution, each $\mathcal{S}_i$ is constructed by Algorithm 4, which adds the nodes to set $\mathcal{S}_i$ (denoted by temporary variable R) one by

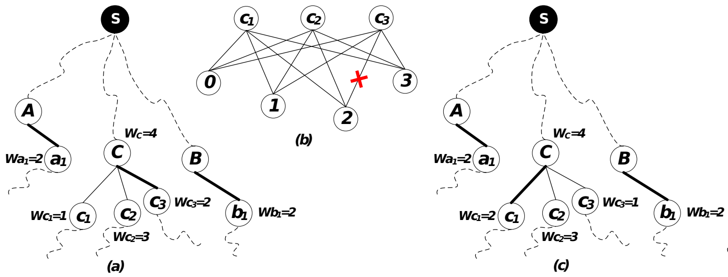


Fig. 3: Scheduling and interference resolution in part of an example aggregation tree: (a) scheduling under one-hop interference model, (b) matching graph modification for parent C, (c) scheduling under SINR

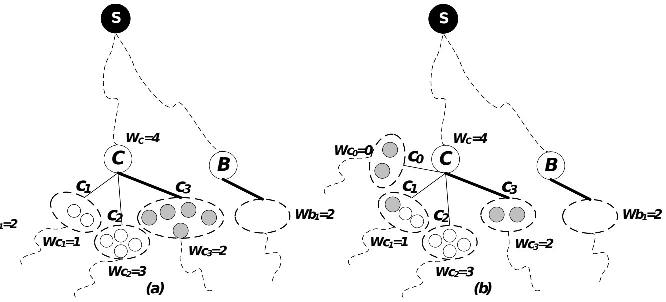


Fig. 4: Regrouping for interference resolution in meta-nodes in part of an example aggregation tree: (a) before resolution, (b) after resolution

one (Alg. 4, Line 4) then checks whether or not the added node causes interference. In case of interference, the added node is removed and the algorithm tries to find an alternative one. Therefore, $\mathcal{S}_i$ cannot contain the interfering nodes, and hence, the final solution is interference-free. ■

IV. SIC Solution

As mentioned earlier, SIC is a subcategory of multi-packet reception techniques. Thus, for each parent, we partition its children into a few groups of concurrent transmitters—called *meta-nodes* as shown in Fig. 1. By reloading and visualizing a higher-level hierarchy (in an overlay view), our problem becomes the scheduling of meta-nodes. In the new setting, we extend our solution under SINR to wait assignment and interference resolution of meta-nodes. Note, however, that to maximize the utilization of transmission capacity in each slot during a random walk on the Markov chain, a parent may regroup the set of its children if an interfering meta-node can partially succeed in concurrent transmissions.

Different grouping strategies for construction of meta-nodes can be employed, although it is beyond the scope of this paper. Here, for simplicity, we use an ID-based grouping, where we sort and partition the children of a parent in a descending order of their IDs while satisfying the SIC constraint in (7). After grouping the nodes, like in the SINR solution, the heuristic is executed in two steps: (1) running *BASIC* on a tree of meta-nodes, where the weight $X[m, W_m]$ in matching graph of a meta-node m is given by the sum of QoA of its nodes under delay constraint W_m, and (2) resolving the globally-imposed interferences using a random walk on a Markov chain—in which each state corresponds to a possible permutation of the interfering meta-nodes—and matching graph modification, on a slot-by-slot basis. However, as mentioned earlier, we do not remove the link between an interfering meta-node and current slot in its associated matching graph if it can *partly* succeed in concurrently transmitting its data. Instead, we launch regrouping in which the nodes (in an ascending order of their IDs) are removed from the interfering meta-node until an interference-free solution is found. Then, the removed nodes are pushed into the other non-scheduled meta-nodes of the same parent as much as possible while the remaining ones make a new group (meta-node). Finally, after interference resolution in each slot, we run *BASIC* on the subtree of each parent with a modified matching graph to update its participating successors and their waiting times.

To give an example, without loss of generality, assume that c_1, c_2, and c_3 are three possible meta-nodes scheduled by *BASIC* without consideration of global interference and $\{b_1, c_3\}$ is the current permutation of meta-nodes in the underlying Markov state. Here, transmission of c_3 along with set $\{b_1\}$ is a case of interference under SIC model. To fully utilize waiting 2, Fig. 4 shows how participants of c_3 (see gray nodes in the figure) starts to leave this meta-node, one by one, in ascending order of their IDs until an interference-free solution in c_3 is obtained. As evident form the figure, the removed nodes are partly pushed to already existing (c_1) and new (c_0) meta-nodes, while those already scheduled meta-nodes (c_2) are left untouched.

V. Complexity Analysis of Algorithms

In this section, we analyze the complexity of our proposed SINR and SIC algorithms. Let h be the height of the aggregation tree, D the deadline imposed by the sink, N the total number of sensor nodes, and T the number of rounds walking on the Markov chain. L is the maximum number of concurrent transmitters in each permutation (note that, in practice, $L << N$) and the in-degree of each node in the tree is bounded by k. Moreover, $k' < k$ and $L' < L$ are the degree of tree of meta-nodes and the maximum number of concurrent meta-nodes in each permutation, respectively.

Theorem 4: The complexities of our SINR and SIC algorithms are $O(DTLk(hk^2(D+k)\log k + L^2))$ and $O(DTL'k'(hk'^2(D+k')\log k' + L'^2k'^4))$, respectively.

Proof: For SINR, let A and B be the cost of Algorithm 4 and *BASIC*, respectively. Each iteration of "for" loop (with total D iterations) in Algorithm 1 includes T rounds of Markov process in Algorithm 3. Therefore, the total complexity of Algorithm 1 can be written as $O(B + D(TA + B)) = O(D(TA + B))$. In Algorithm 4 and each iteration of the "while" loop, we need to check the feasibility of set of concurrent transmitters R and if the check fails, we require to solve an MWM problem. Solving an MWM problem in a bipartite graph is also a part of the *BASIC* algorithm and therefore in the worst case is of $O(B)$. Thus, Algorithm 1 is of $O(DTA)$ time complexity. Since $|R| \in O(L)$, the cost of checking is $O(L^2)$. Moreover, the "while" loop iterates at most k rounds. Therefore, we get the cost of the loop as

$O(k(B+L^2))$. By multiplying at the number of iterations of the "for" loop (which is L), we obtain the total cost of Algorithm 4 as $O(Lk(B+L^2))$. Finally, as B (cost of *BASIC*) is $O(hk^2(D+k)\log k)$ [8], the total cost of Algorithm 1 can be expressed as $O(DTLk(hk^2(D+k)\log k+L^2))$.

For SIC, before running *BASIC*, the initial step is the group formation. Here, the children of each parent in the tree are sorted in $O(k \log k)$ and then partitioned to meta-nodes under SIC in $O(k^2)$. However, the group formation complexity is dominated by those of the other phases. Thus, we still follow the same order of complexity as the SINR solution, i.e., $O(DTA)$, but with a different complexity for A. Actually, $O(Lk(B+L^2))$ in SINR is here replaced by $O(L'k'(B+(k'+k'^2)(L'k')^2))$, where term $(k'+k'^2)(L'k')^2$ is the cost of interference checking and resolution while also regrouping the meta-nodes to maximize the utilization of transmission capacity. Finally, the total cost of Algorithm 1 under SIC can be expressed as $O(DTL'k'(hk'^2(D+k')\log k'+(k'+k'^2)(L'k')^2)) = O(DTL'k'(hk'^2(D+k')\log k'+L'^2k'^4))$. ■

VI. EVALUATION

In this section, we evaluate the performance of different scheduling algorithms in a random tree-based WSN of N sensor nodes uniformly deployed in a $100m \times 100m$ region. The sink is located at the top-center of the region and must meet a deadline of $D = 2, 5, 10, 15$ time slots for data aggregation. We let all nodes serve as sources to provide a better understanding of QoA, i.e., the number of participating sources in data aggregation. Using the same setting in [14], we consider a physical interference (SINR) model with ambient noise power $N_0 = 0.1$, path loss exponent $\alpha = 2.5$, SINR threshold $\beta = 1$, and identical transmission power of $P = 15$ for network nodes. Moreover, as in [1], we let $\alpha_m = 0.2$ and $\beta_m = 2$ in the Markov approximation framework and report the results after $T = 200$ rounds. Due to the space limit, we omit the detailed results on how the above SINR and Markov parameters affect the number of transmissions. Instead, our objective is to show how the underlying deadline and interference constraints hinder participation of all nodes in data aggregation for different algorithms. Each data point of the figures corresponds to the average of 50 runs with different random topologies. The following four algorithms are compared in terms of their QoA under various deadlines and network sizes.

- *BASIC* [8] provides the optimal solution and maximizes QoA under the one-hop interference model. This result gives an upper bound of our solution under SINR.
- *SINR* is our proposed algorithm that maximizes QoA under the SINR model.
- *SINR-Simple* is our SINR algorithm, but without the replacement search phase, thus under-utilizing the power of our solution approach.
- *SIC* is our proposed algorithm that maximizes QoA under the SIC model.

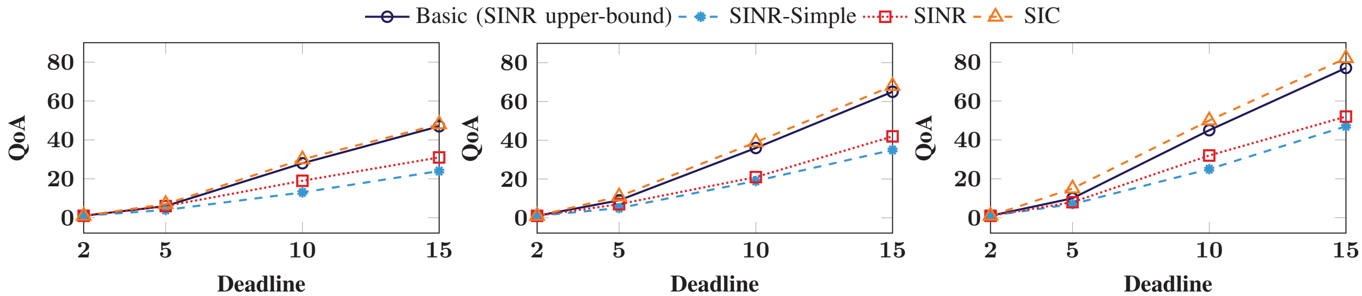
We first study the effect of deadline on QoA. Fig. 5 shows how QoA improves as the deadline increases. That is, by increasing the sink deadline, more source nodes will have the opportunity to participate in the aggregation process. As evident from the figure, applying globally-imposed interferences can significantly decrease QoA under the SINR model compared to the one-hop interference model. However, by using multi-packet reception in our *SIC* algorithm, we not only outperform the *SINR* algorithm on average by 44% but also go over the performance of *BASIC*. Moreover, the figure demonstrates how using the replacement search policy and careful modification of matching graphs in our algorithm can improve QoA. In sparser networks, *SINR* outperforms *SINR-Simple* significantly, e.g., by an average of 31% in a network of 50 sensor nodes as shown in Fig. 5(a). However, as density increases, the results show a smaller difference between *SINR* and *SINR-Simple*. The reason is that extra interferences incurred by the new replacing transmitters reduce the opportunity of concurrent transmissions in other participants. This is an interesting result in terms of scalability, meaning that we can remove the replacement search phase in high density networks and still achieve the maximum QoA while running a lower complexity algorithm.

Moreover, Fig. 5 shows the effect of network size ($N =$ 50, 100, 150) on QoA. As shown in the figure, increasing network density can improve QoA. However, the ratio of participation of nodes is observed to decrease (from almost all nodes in $N =$ 50 to half of nodes in $N =$ 150) while network density grows. The reason is that interference limits the number of concurrent transmissions in the network. Hence, increasing density alone would not always benefit the QoA in WSNs.

VII. RELATED WORK

The choice of the interference model is of fundamental significance to any study of wireless networks. Although a graph-based (i.e., hop- and range-based, and protocol) interference model provides a useful abstraction, it is an oversimplification of the physical reality. Instead, the physical interference (a.k.a. SINR) model represents the real interference behavior more accurately, since the success of a packet reception depends on all concurrently scheduled transmissions in the network [6], [21]. The choice is much more challenging when dealing with real-time data aggregation classified into the two main categories in WSNs (i) minimum latency aggregation scheduling (MLAS), and (ii) deadline-constrained aggregation scheduling. The first category has already been studied extensively under different interference models [13], [14], [19]. The current study focuses on the second category, where the basic question is how much information can be aggregated from a deadline-constrained WSN. As mentioned earlier, no solution for MLAS can be employed to solve this problem.

Hariharan and Shroff [8] first introduced this problem and presented an optimal solution that involves a localized maximum weighted matching problem at each hop. In [9], this problem was extended to tree-based WSNs with unreliable links, shown to be NP-hard in the strong sense, and solved using a dynamic programming framework. Considering the effect of data redundancy due to spatial correlation, Alinia



(a) $N = 50$ **(b)** $N = 100$ **(c)** $N = 150$

Fig. 5: QoA vs. Deadline for four scheduling algorithms under different network sizes

et al. [2] extended the problem to consider both the number of source participants and their spatial dispersion as the underlying merit factors for maximizing QoA. However, all above studies solve the problem under the simplest (i.e., one-hop) interference model, because the inclusion of global interferences—as done in this paper—substantially increases the difficulty of the problem.

VIII. Discussion and Future Direction

QoA definition: We defined QoA as the number of source nodes whose data have been accounted for at the sink within an imposed deadline [8]. However, our optimization framework is general enough to consider QoA in a number of different ways by modifying $X[.,.]$ in our formulation. Maximizing the total priority of packets if each source node is assigned a priority, maximizing aggregation accuracy if each source's observation is associated with a particular confidence index, and maximizing both the number of participating sources and their spatial dispersion are examples we can study in future.

Distributed implementation: The current study relies on centralized interference management. Although our approach makes an important contribution by maximizing QoA under SINR/SIC constraints and is promising under emerging architectures like SDN-based WSNs [11], [15], [16], in future we would like to develop a fully distributed solution which is more practical for "large-scale" WSN deployments.

IX. Conclusion

To the best of our knowledge, this is the first to address the problem of interference-free scheduling under the physical interference (SINR) model for maximizing quality of aggregation (QoA) in deadline-constrained WSNs. The formulated problem was first proved to be NP-complete. We then proposed a suboptimal scheduling algorithm which uses a Markov approximation framework and modifies matching graphs to handle the globally-imposed interferences. The problem (and solution) was then coupled with successive interference cancellation (SIC) to improve QoA. We also derived the theoretical upper bounds on QoA under the SINR and SIC models. Finally, our evaluation results demonstrated the effectiveness of the proposed scheduling approaches.

References

[1] B. Alinia, M. H. Hajiesmaili, and A. Khonsari, "On the construction of maximum-quality aggregation trees in deadline-constrained wsns," in *INFOCOM'15*, 2015, pp. 226–234.

[2] B. Alinia, H. Yousefi, M. S. Talebi, and A. Khonsari, "Maximizing quality of aggregation in delay-constrained wireless sensor networks," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2084–2087, 2013.

[3] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE TIT*, vol. 59, no. 10, pp. 6301–6327, 2013.

[4] X. Chen, X. Hu, and J. Zhu, "Minimum data aggregation time problem in wireless sensor networks," in *MSN'05*, 2005, pp. 133–142.

[5] D. Gong and Y. Yang, "Low-latency SINR-based data gathering in wireless sensor networks," in *INFOCOM'13*, 2013, pp. 1941–1949.

[6] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric SINR," in *MobiHoc'07*, 2007, pp. 100–109.

[7] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *MobiCom'08*, 2008, pp. 339–350.

[8] S. Hariharan and N. Shroff, "Maximizing aggregated information in sensor networks under deadline constraints," *IEEE TAC*, vol. 56, no. 10, pp. 2369–2380, 2011.

[9] S. Hariharan, Z. Zheng, and N. B. Shroff, "Maximizing information in unreliable sensor networks under deadline and energy constraints," *IEEE TAC*, vol. 58, no. 6, pp. 1416–1429, 2013.

[10] S. Ji, R. Beyah, and Y. Li, "Continuous data collection capacity of wireless sensor networks under physical interference model," in *MASS'11*, 2011, pp. 222–231.

[11] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[12] H. Li, C. Wu, D. Yu, Q. S. Hua, and F. C. M. Lau, "Aggregation latency-energy tradeoff in wireless sensor networks with successive interference cancellation," *IEEE TPDS*, vol. 24, no. 11, pp. 2160–2170, 2013.

[13] H. Li, C. Wu, Q.-S. Hua, and F. C. M. Lau, "Latency-minimizing data aggregation in wireless sensor networks under physical interference model," *Ad Hoc Networks*, vol. 12, pp. 52–68, 2014.

[14] X. Y. Li, X. Xu, S. Wang, S. Tang, G. Dai, J. Zhao, and Y. Qi, "Efficient data aggregation in multi-hop wireless sensor networks under physical interference model," in *MASS'09*, 2009, pp. 353–362.

[15] T. Luo, H. P. Tan, and T. Q. S. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.

[16] M. Moradi, Y. Lin, Z. Mao, S. Sen, and O. Spatscheck, "SoftBox: A customizable, low-latency, and scalable 5G core network architecture," *IEEE Journal on Selected Areas in Communications*, 2018.

[17] G. Wang, Q.-S. Hua, and Y. Wang, "Minimum latency aggregation scheduling for arbitrary tree topologies under the SINR model," in *ADHOC-NOW*. Springer, 2012, pp. 139–152.

[18] S. P. Weber, J. G. Andrews, X. Yang, and G. de Veciana, "Transmission capacity of wireless ad hoc networks with successive interference cancellation," *IEEE TIT*, vol. 53, no. 8, pp. 2799–2814, 2007.

[19] H. Yousefi, M. Malekimajd, M. Ashouri, and A. Movaghar, "Fast aggregation scheduling in wireless sensor networks," *IEEE TWC*, vol. 14, no. 6, pp. 3402–3414, 2015.

[20] S. Zhang, Z. Shao, M. Chen, and L. Jiang, "Optimal distributed p2p streaming under node degree bounds," *IEEE/ACM TON*, vol. 22, no. 3, pp. 717–730, 2014.

[21] Y. Zhou, X. Y. Li, M. Liu, X. Mao, S. Tang, and Z. Li, "Throughput optimizing localized link scheduling for multihop wireless networks under physical interference model," *IEEE TPDS*, vol. 25, no. 10, pp. 2708–2720, 2014.