Ad Hoc & Sensor Wireless Networks, Vol. 26, pp. 233–257
Reprints available directly from the publisher
Photocopying permitted by license only

Minimum Latency Aggregation Scheduling in Wireless Sensor Networks with Successive Interference Cancellation

SHILIANG XIAO[1,2,3,*], LEBING PAN[1,2,3], YUNZHOU QIU[1,2] AND XIAOBING YUAN[1,2]

[1]*Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China, 200050*
[2]*Science and Technology on Microsystem Laboratory, Shanghai, China, 200050*
[3]*University of Chinese Academy of Sciences, Beijing, China, 100049*

Received: April 27, 2013. Accepted July 17, 2013.

We study the minimum latency aggregation scheduling (MLAS) problem in wireless sensor networks (WSNs) under the physical interference model combined with successive interference cancellation (SIC). Recognized as a powerful technique of multi-packet reception (MPR) at the physical layer, SIC allows a receiver to decode several arriving signals at the same time as long as these signals differ in strength significantly, resulting in increased transmission throughput and reduced latency, potentially. We first formulate the problem of MLAS with SIC and prove its NP-hardness. To resolve it effectively, we then propose two heuristic polynomial-time scheduling algorithms, namely *first-fit aggregation scheduling* (F2AS) and *shortest-fit aggregation scheduling* (SFAS), through fully exploiting the new transmission opportunities offered by SIC. Both algorithms are greedy ones that try to achieve maximal possible links scheduled in each time slot, thereby minimizing the total time consumed by the data aggregation task. Theoretical analysis indicates that the schedule results of F2AS and SFAS are both interference-free. Simulations show that F2AS and SFAS outperform state-of-the-art scheduling algorithms designed for MLAS under the physical interference model without SIC in terms of latency under various network configurations.

Keywords: Data aggregation, scheduling, physical interference model, successive interference cancellation, wireless sensor networks

* E-mail: shliangxiao@gmail.com

1 INTRODUCTION

In many applications of wireless sensor networks (WSNs) such as environment monitoring, military surveillance and industrial control, a set of small-sized and low-powered sensor nodes spreading over the targeted area are desired to sense the surroundings and report the sensory data to the end user (i.e. the sink) for further analysis. The information flow from a number of nodes toward a single sink is termed as *many-to-one communication* [5], which is a fundamental transmission pattern in WSNs. During the process of many-to-one communication, data may be compressed at intermediate nodes along the path to the sink, leading to the *data aggregation* operation [16]. In contrast to raw data gathering, data aggregation reduces transmissions and saves energy notably through fusing data from different nodes with some aggregation functions such as MAX, MEAN, AVG and etc [5].

For delay-sensitive applications, the time of utilizing the data is critical and out-of-date data is not only useless but sometimes even harmful. At such circumstances, low-latency data aggregation operation is preferred. Here, the latency of data aggregation is the duration from the time when the first node transmits to the time when the sink receives the final aggregated data from all nodes. Due to the *broadcast nature* of wireless channel, wireless transmission suffers severe interference from concurrently active links, serving as the main cause of data aggregation latency [2]. To deal with wireless interference appropriately, the minimum latency aggregation scheduling (MLAS) problem is proposed and has been extensively studied in the literature. MLAS aims to find an interference-free TDMA schedule for data aggregation, while minimizing the total latency [2]. The main technique obstacle of MLAS is how to model the wireless interference accurately. Throughout the literature, two kinds of interference models are widely adopted, namely *protocol interference model* and *physical interference model* [8].

MLAS under the protocol model has been well studied in [12, 14, 17, 20, 26, 30, 32], where interference is assumed to exist between any two links that are within a certain range from each other. With such an assumption, conflicting graphs are constructed to describe interference relationship within the network and algorithms based on graph-coloring are raised to provide collision-free schedules. While the protocol model acts as a useful abstraction of WSNs, it does not reflect the practical interference of wireless transmission since it simply neglects the interference from nodes beyond a certain range. In fact, though the interference from a single transmitter may be relatively small, the accumulated interference from multiple transmitters can be high enough to corrupt a wireless transmission [31]. To capture wireless interference more realistically and accurately, the physical model is adopted, under which MLAS has attracted much concentration in recent years in [1, 4, 11, 18, 27, 31]. In the physical model, the superimposed interference from all active links are taken into account and a signal can be decoded successfully if and only if the *signal-to-interference-plus-noise-ratio* (SINR) is above a hardware-defined threshold. That means the receiver tries to decode the intended signal while treating all the other ones as interference and noise. However, up-to-date development at the physical layer shows that by using successive interference cancellation (SIC), an effective technique of multi-packet reception, a receiver may be able to decode several signals at the same time using an iterative approach [9]. To be more specific, in each iteration the strongest signal is decoded while treating the remaining weaker ones as interference. Then it will be removed from the composite signal if successfully decoded according to the SINR requirement. In the subsequent iteration, the next strongest signal is decoded and the process continues until either the intended signal is decoded successfully or the decoding of some signal is failed. Theoretical and empirical studies have shown that with SIC, the performance of wireless networks can be improved significantly [13, 21, 28]. Since the receivers are allowed to decode multiple signals concurrently and more transmissions can be scheduled at the same time, the latency of data aggregation is expected to be reduced notably provided that SIC is coupled with the physical model.

In this paper, we investigate MLAS under the physical interference model with the help of successive interference cancellation. Such work is nontrivial in order to accurately measure the practical benefit gained by applying SIC in typical wireless applications. Specifically, we make the contributions summarized as below.

1. We formulate the problem of MLAS under the physical interference model with successive interference cancellation (termed MLAS-SIC) by identifying the fundamental constraints involved in solving this problem. With regard to the complexity of MLAS-SIC, we theoretically show that it is NP-hard in general.
2. We propose two heuristic scheduling algorithms, i.e. first-fit aggregation scheduling (F2AS) and shortest-fit aggregation scheduling (SFAS) for MLAS-SIC. F2AS randomly picks a transmission link for scheduling while SFAS always prefers the shortest one among the set of feasible links. Both algorithms are greedy ones that try to schedule as many links as possible during each time slot by fully exploiting the new transmission opportunities offered by SIC.
3. We theoretically prove that the schedule results of F2AS and SFAS are both interference-free. We also provide computational complexity analysis and show that both algorithms can be accomplished in polynomial time.

4. With extensive simulations, we show that F2AS and SFAS outperform two state-of-the-art scheduling algorithms designed for the physical model without SIC in terms of aggregation latency under various network configurations.

Our work indicates that SIC is indeed an effective way to alleviate wireless interference and data aggregation can benefit from SIC notably, provided that SIC is efficiently exploited and the scheduling algorithms are carefully designed. The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the network model and problem definition. The scheduling algorithms and the corresponding analysis are provided in Section 4 and Section 5, respectively. The simulation results are presented in Section 6 and finally in Section 7, we conclude the paper and give some directions for further research.

2 RELATED WORK

2.1 Minimum Latency Aggregation Scheduling

There has been much work on MLAS with various interference models in the literature. Under the protocol model, Chen *et al.* [2] prove that MLAS is NP-hard and propose a scheduling algorithm with a latency bound of $(\Delta - 1) \cdot R$ time slots, where Δ is the maximum node degree and R is the network radius. Afterwards, Huang *et al.* [12] and Wan *et al.* [26] propose improved algorithms with upper bounds of $23R + \Delta - 18$ and $15R + \Delta - 4$, respectively. Yu *et al.* [32] propose the first distributed scheduling algorithm with a latency bound of $48R + 6\Delta + 16$, which is further improved by Li *et al.* [20] and Xu *et al.* [30].

Under the more accurate physical model, MLAS has attracted much attention in recent years. Based on the so called *grid partition and coloring* method, Xu *et al.* [31] propose an efficient scheduling algorithm with a latency bound of $O(\Delta + R)$ for random networks with uniform power assignment. A drawback of [31] is that it cannot achieve the same approximation performance when applied to arbitrary networks. Assuming that the maximum power level of nodes is infinite, Li *et al.* [18] propose a scheduling algorithm for arbitrarily deployed networks with a latency bound of $O(\rho)$, where ρ is the link length diversity defined as the logarithm of the ratio between the lengths of the longest link and the shortest link. For the case where multiple power levels are available and the maximum power level is bounded, Min *et al.* [1] propose a scheduling algorithm using a similar method with Xu *et al.*'s [31]. Another contribution of [1] is the derivation of the NP-hardness of MLAS under the physical interference model, which has not been obtained before.

Despite of the distinction of the underlying interference models, all above algorithms share a common idea of *interference avoidance*, which achieves interference-free transmissions by removing any overlap among the transmitting signals (the root of interference). Although practical, interference avoidance can not offer a performance close to the network information theoretical limit, which can only be achieved through some kind of *interference exploitation* approach such as SIC [24].

2.2 Successive Interference Cancellations

Successive interference cancellation (SIC) is a type of multi-packet reception technique in wireless communication. Weber *et al.* [29] investigate the improvement of transmission capacity obtainable with SIC in ad hoc wireless networks. They successfully develop closed-form upper bounds and easily computable lower bounds of the transmission capacity in wireless networks with SIC receivers, for both perfect and imperfect SIC. In [9], SIC is implemented at the physical layer of IEEE 802.15.4 standard, where collided signals have to differ significantly in power in order to be decoded. Wang *et al.* [28] present a polynomial-time heuristic algorithm to approximate the optimal throughput in ad hoc networks with joint routing and scheduling with SIC. Later on, Lv *et al.* [21] propose a novel notation of simultaneous graph to characterize the effect of SIC on link dependence due to interference, and present a greedy scheduling scheme based on maximal independent set (MIS). Recently, Ghaderi *et al.* [6] study the problem of uplink scheduling in wireless networks with SIC. Specifically, they formulate a series of scheduling problems like maximum throughput scheduling (MTS), proportional fairness scheduling (PFS) and scheduling with minimum SINR (MinSINR), and for each problem, they either propose a polynomial time algorithm or show that the problem is NP-hard. In another recently online published work [23], Sen *et al.* explore the extent of throughput gains possible with SIC from the perspective of MAC layer, for the first time. Through rigorous analysis and extensive simulations, they show that interfering one-to-one transmissions benefit less from SIC than scenarios with many-to-one transmissions, such as when clients upload data to a common access point.

Anyhow, we can see that though SIC has long been recognized as a powerful physical layer technique by the communication community, its potential in improving the performance of algorithms and protocols at the link and network layers is largely unexplored. Fortunately, research in this area is emerging nowadays and hoped to increase in the future. Our work in this paper, aiming at improving the latency performance of data aggregation in WSNs with SIC, is expected to serve as a valuable supplement for the research in this promising area.

2.3 Discussions of Reference [19]

Recently, aware of the potential brought by SIC, Li *et al.* [19] online publish a paper focused on optimizing latency and energy consumption of data aggregation in WSNs when SIC is available at the physical layer. Specifically, they present theoretical lower bounds on both latency and energy as well as their tradeoff, and introduce an efficient approximation algorithm that can achieve asymptotical optimum in both aggregation latency and latency-energy tradeoff. However, there exist several differences distinguishing our work from [19]. First, the problem addressed by [19] is coupled with accurate power control (where oblivious power control is adopted) while in this paper we concentrate on the scheduling problem itself and consider uniform power assignment [8], which is simple yet widely used in practical systems. Second, the objective in [19] is joint optimization of both latency and energy assumption and because of this objective and the consideration of reducing energy and facilitating implementation, during the data aggregation process the opportunities offered by SIC are only partially exploited in [19] compared with our work here, as will be verified in the simulation section. Third but not the least, the authors make an important assumption about the network topology by which no two nodes are permitted to be situated within $1m$ from each other, while in this paper, we do not restrict the network topology at all. In other words, our scheduling algorithms adapt to arbitrary networks.

3 MODEL AND PROBLEM

3.1 Network Model

We consider a set of nodes (denoted by $\mathcal{V}$) deployed arbitrarily in a 2D plane where $v_s \in \mathcal{V}$ is the sink node. Each node is equipped with an omnidirectional antenna and can send (receive) data to (from) all directions. For a transmission link l in the network, we denote the transmitter and receiver of l by $x(l)$ and $r(l)$, respectively. Let $d(u, v)$ denote the Euclidean distance between nodes u and v. For link l, its length is denoted by $d(l) = d(x(l), r(l))$. For links l' and l, the distance from l' to l is denoted by $d(l', l) = d(x(l'), r(l))$.

Under the physical interference model [8], we assume all nodes have fixed and uniform transmission power P, and for a transmission link l, we denote the received power at $r(l)$ of the signal transmitted from $x(l)$ by $P(l) = \frac{P}{d^{\alpha}(l)}$. Here α $(2 < \alpha \leq 6)$ is the path-loss exponent whose exact value depends on external conditions of the wireless medium such as obstacles, humidity, and etc. The transmission of l is successful if and only if the signal-to-interference-plus-noise-radio (SINR) at $r(l)$ is above a hardware-defined threshold β $(\beta > 1)$:

$$\frac{P(l)}{\xi + \sum_{l' \in \mathcal{S}_t}^{l' \neq l} P(l', l)} \geq \beta \tag{1}$$

Here $\mathcal{S}_t$ is the set of links concurrently active with link l, ξ ($\xi > 0$) is the ambient noise power and $P(l', l)$ is the interference experienced by $r(l)$ coming from the transmitter of link l', i.e. $P(l', l) = \frac{P}{d^{\alpha}(l', l)}$. The maximum transmission range with power P under the physical model can be computed as $r = (\frac{P}{\xi\beta})^{\frac{1}{\alpha}}$. That means r is the maximum possible length of a communication link where the receiver can successfully decode the signal without interference from any other link. We use $G(\mathcal{V}, \mathcal{E})$ to denote the network, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges whose lengths are not longer than r.

Note that under the physical interference model, we only intend to decode the desired signal and all the other concurrent ones are treated as interference. If one of the concurrent signals is stronger than the desired one, then we are surely unable to decode the desired signal according to (1). However, with the help of SIC, we may decode the relatively smaller signal. When SIC is employed, a receiver is able to decode multiple signals simultaneously provided that these signals differ in strength significantly. As we have described, SIC performs in an iterative fashion and the current strongest signal is decoded in each iteration while treating the remaining weaker ones as interference. Then it will be removed from the composite signal if successfully decoded according to (1). This process continues until either the desired signal is decoded successfully or the decoding of some signal is failed. At such circumstances, the transmission of link l is successful if and only if (a) any stronger active link l' (i.e. $P(l', l) > P(l)$) satisfies:

$$\frac{P(l', l)}{\sum_{P(l'', l) \leq P(l', l)}^{l'' \in \mathcal{S}_t, l'' \neq l'} P(l'', l) + \xi} \geq \beta \tag{2}$$

(b) link l satisfies:

$$\frac{P(l)}{\sum_{P(l', l) \leq P(l)}^{l' \in \mathcal{S}_t, l' \neq l} P(l', l) + \xi} \geq \beta \tag{3}$$

3.2 Problem Definition

We now formulate the minimum latency aggregation scheduling problem under the physical interference model with successive interference cancellation (termed MLAS-SIC), formally. We consider a synchronous message passing model in which time is divided into slots and each slot is long enough for a node to transmit or receive a message. Given that each node in $\mathcal{V}$ has one message to send and $\mathcal{A}$, $\mathcal{B}$ are two disjoint subsets of $\mathcal{V}$, data can be aggregated from $\mathcal{A}$ to $\mathcal{B}$ in one time slot if and only if all signals transmitted by nodes in $\mathcal{A}$ can be decoded successfully with the help of SIC by some nodes in $\mathcal{B}$. Then a valid aggregation schedule with delay K for all nodes in $\mathcal{V}$ can be defined as a sequence of sender sets $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_K\}$ satisfying the following constraints:

1. $\bigcup_{i=1}^{K} \mathcal{S}_i = \mathcal{V} \backslash \{v_s\}$
2. $\mathcal{S}_i \bigcap \mathcal{S}_j =$NULL , $\forall i \neq j$
3. For all i $(i = 1, 2, \cdots, K)$, data can be aggregated from $\mathcal{S}_i$ to $\mathcal{V} \backslash \bigcup_{j=1}^{i} \mathcal{S}_j$ in one time slot successfully, meeting the inequalities (2) and (3).

The first constraint ensures that all data will be finally aggregated to v_s. The second constraint ensures that each node send data only once. The third constraint ensures that all transmissions in each time slot are successful, i.e. all the signals of the active links are decoded successfully by the intended receivers using SIC. MLAS-SIC seeks a valid schedule $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_K\}$ such that the latency K is minimized. In terms of MLAS-SIC, we have the following theorem indicating its complexity.

Theorem 1. *The problem of MLAS-SIC is NP-hard.*

Proof. Please refer to Appendix A for detailed proof.

4 ALGORITHM DESIGN

In this section, we present detailed description about the proposed algorithms F2AS and SFAS designed for solving the MLAS-SIC problem. We will provide the corresponding algorithm analysis in the next section.

4.1 Technical Preliminaries

For briefness of clarification and ease of analysis, we first introduce some important notations that will be used in the detail of our algorithms throughout the paper. We define *relative interference* (RI) caused by the transmitter of link l' to the receiver of link l as

$$RI(l', l) = \begin{cases} 0 & l' = l \quad \text{(4a)} \\ \infty & d(l', l) = 0 \text{ and } l' \neq l \quad \text{(4b)} \\ c(l) \cdot \dfrac{d^{\alpha}(l)}{d^{\alpha}(l', l)} & \text{others} \quad \text{(4c)} \end{cases}$$

Here $c(l) = \frac{P\beta}{P - d^{\alpha}(l) \cdot \beta\xi}$ is a factor depending only on the property of link l itself and some universal constants. RI is also referred to as *affectance*, which can be found in [7, 22, 25] with various forms.Let $\mathcal{S}_t$ be the set of simultaneous scheduled links with link l. Likewise, we have RI from $\mathcal{S}_t$ on l as $RI(\mathcal{S}_t, l) = \sum_{l' \in \mathcal{S}_t} RI(l', l)$. By the notation of RI, it can be easily verified that the following proposition holds.

Proposition 1. *The inequality (1) of the physical interference model is equivalent to* $RI(\mathcal{S}_t, l) \leq 1$.

With SIC, in order to decode the signal of l, we must first decode all stronger signals at $r(l)$ coming from the transmitters of the links in $\mathcal{S}_t$. We define such a stronger link as an *associated link* (AL) of l and all the associated links of l among $\mathcal{S}_t$ make up an *associated link set* (ALS), represented as $ALS(l, \mathcal{S}_t) = \{l' | l' \in \mathcal{S}_t, P(l', l) > P(l)\}$. Likewise, we have $ALS(l', l, \mathcal{S}_t) = \{l'' | l'' \in \mathcal{S}_t, P(l'', l) > P(l', l) > P(l)\}$, which is the set of links that must be successfully decoded before l'. Based on the above notations,we can achieve the following proposition straightly.

Proposition 2. *The inequalities (2) and (3) of the physical interference model combined with SIC are equivalent to the condition as follows: for each link* $l' \in ALS(l, \mathcal{S}_t) \cup \{l\}$, $RI(l', l) \geq \frac{\beta}{\beta+1} \cdot RI(\mathcal{S}_t \backslash ALS(l', l, \mathcal{S}_t), l) + \frac{c(l)-\beta}{\beta+1}$ *holds.*

4.2 Algorithm Outline

Our scheduling algorithms perform in a layer-by-layer fashion. We first divide the nodes in $\mathcal{V}$ into a series of subsets according to their hop distances to the sink v_s. Each subset of nodes is referred to as a layer. For each layer, we schedule the corresponding nodes to send their data to the nodes in the upper layer, where the data aggregation operation is carried out. From the bottom layer till the top layer, we make sure that all the data in the network are finally aggregated to v_s. Specifically, our scheduling algorithms comprise the following steps.

1. Apply a *breadth-first-search* (BFS) algorithm [3] on $G(\mathcal{V}, \mathcal{E})$ to construct a *fat tree* rooted at v_s. Here, the fat tree is the union of all shortest path trees (SPTs), where branches from v_s to each node are paths with the least hop count [3]. We remove any edge that is between two nodes at the same layer. Figure 1(a) shows a sample of a fat tree.
2. Let R be the height of the fat tree (which is also the radius of G). For i ($i = R, R-1, \cdots, 1$), we construct a series of bipartite graphs $G_b(\mathcal{V}(i), \mathcal{V}(i-1), \mathcal{E}(i))$, where $\mathcal{V}(i)$ consists of nodes in $\mathcal{V}$ that are i-hop from v_s and $\mathcal{E}(i)$ consists of edges between nodes in $\mathcal{V}(i)$ and nodes in $\mathcal{V}(i-1)$. Figure 1(b) shows a sample of a bipartite graph.

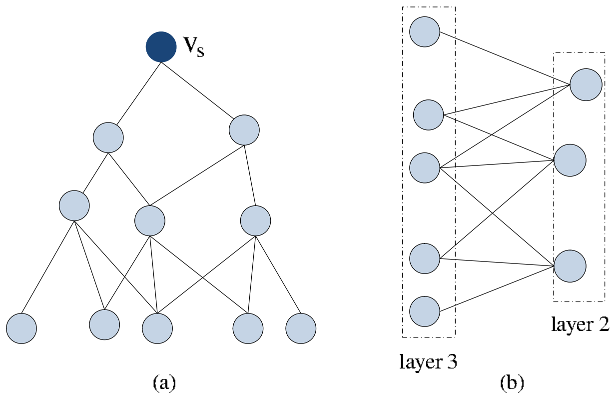


FIGURE 1
(a) A fat tree with height $R = 3$. (b) A bipartite graph made up of nodes at layer 2 and layer 3 of the fat tree.

3. Given each $G_b(\mathcal{V}(i), \mathcal{V}(i-1), \mathcal{E}(i))$ as input, we use F2AS (SFAS) to aggregate the data from all nodes in $\mathcal{V}(i)$ to some nodes in $\mathcal{V}(i-1)$. By executing F2AS (SFAS) up to R times, we make sure that v_s receive the final aggregated data from all the nodes in the tree.

Now we present detailed description about F2AS and SFAS, which serves as the key ingredient of our solutions.

4.3 First-fit Aggregation Scheduling

First-fit aggregation scheduling (F2AS) is a simple heuristic algorithm which greedily schedules links through continuously checking the opportunities of simultaneous transmissions. The pseudo-code of F2AS is presented in Algorithm 1. F2AS takes a bipartite graph $G_b(\mathcal{U}, \mathcal{W}, \mathcal{L})$ as input, where $\mathcal{U}$ is the set of senders, $\mathcal{W}$ is the set of receivers and $\mathcal{L}$ is the set of all possible edges (links) between nodes in $\mathcal{U}$ and $\mathcal{W}$. The links in $\mathcal{L}$ are scheduled as follows. First we randomly pick a link l from $\mathcal{L}$, and add it to the scheduled set $\mathcal{S}_t$. Then for each of the remaining link l' in $\mathcal{L}$, we will determine whether it violates the interference checking rules for F2AS by calling the subroutine described in Algorithm 2. In specific, Algorithm 2 implements the interference checking process for F2AS by directly checking whether adding l' to $\mathcal{S}_t$ will violate the decoding conditions for all the links in $\mathcal{S}_t \cup \{l\}$ based on Proposition 2 stated in Section 4.1. For the link that can not pass the interference checking process, we will add it to a temporary set $\mathcal{M}$. In contrast, for the link that are able to pass the interference checking process, we will add it to $\mathcal{S}_t$. Generally, every time when a link l is scheduled, we will remove all

Algorithm 1: First-fit Aggregation Scheduling

Input: A bipartite graph $G_b(\mathcal{U}, \mathcal{W}, \mathcal{L})$.
Output: A sequence of schedule $\{\mathcal{S}_t\}$. $\mathcal{S}_t$ contains the links that are scheduled in time slot t.

$t \leftarrow 1, \mathcal{S}_t \leftarrow$ NULL, $\mathcal{M} \leftarrow$NULL;
while $\mathcal{L}$ *is not* NULL **do**
  Pick a link l randomly from $\mathcal{L}$, $\mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{l\}$;
  $\mathcal{L} \leftarrow \mathcal{L} \backslash$\{links that are incident with $x(l)$\};
  for *each link* $l' \in \mathcal{L} \backslash \mathcal{M}$ **do**
    if l' *violates the interference checking rules by calling* ***Algorithm 2*** **then**
      $\mathcal{M} \leftarrow \mathcal{M} \cup \{l'\}$;
    else
      $\mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{l'\}$;
      $\mathcal{L} \leftarrow \mathcal{L} \backslash$\{links that are incident with $x(l')$\};
    end
  end
  $t \leftarrow t + 1, \mathcal{M} \leftarrow$NULL;
end

links that are incident with $x(l)$ ($x(l) \in \mathcal{U}$) from $\mathcal{L}$, since a sender only needs to be scheduled once. By checking every remaining link using Algorithm 2 at least once, we make sure that as many links be scheduled in each time slot as possible. Subsequently, we initialize $\mathcal{M}$ to NULL, accumulate t by 1 and repeat the above process until $\mathcal{L}$ becomes NULL, i.e. all senders in $\mathcal{U}$ are scheduled to transmit their data to some nodes in $\mathcal{W}$ successfully.

4.4 Shortest-fit Aggregation Scheduling

Unlike F2AS which randomly picks a link for scheduling, SFAS always prefers the link with the shortest length among all possible choices. As shown in Algorithm 3, at each time slot we pick the shortest unscheduled link, namely l_m, and add it into $\mathcal{S}_t$. For each of the remaining unscheduled links at the current time slot, we will check whether it violates the interference checking rules for SFAS by calling the subroutine described in Algorithm 4. Observe that there are two classes of links that will be considered violating the interference checking rules. The first class contains links whose relative interference (RI) on l_m is beyond a certain threshold $\lambda \cdot c(l_m)$, where λ is a carefully selected parameter determined by α and β, as given in Line 1 of Algorithm 4. The second class contains links whose receivers can not decode the intended signals successfully *to a certain extent* due to the impact of the

Algorithm 2: Interference Checking for F2AS

Input: A schedule $\mathcal{S}$, a link l.
Output: TRUE if l violates the interference checking rules for the scheduled links in $\mathcal{S}$, FALSE otherwise.

for *each link* $l' \in \mathcal{S} \cup \{l\}$ **do**
  for *each link* $l'' \in ALS(l', \mathcal{S}) \cup \{l'\}$ **do**
    if $RI(l'', l') < \frac{\beta}{\beta+1} \cdot RI(\mathcal{S} \backslash ALS(l'', l', \mathcal{S}), l) + \frac{c(l')-\beta}{\beta+1}$ **then**
      return TRUE;
    else
      continue;
    end
  end
end
return FALSE

Algorithm 3: Shortest-fit Aggregation Scheduling

Input: A bipartite graph $G_b(\mathcal{U}, \mathcal{W}, \mathcal{L})$.
Output: A sequence of schedule $\{\mathcal{S}_t\}$. $\mathcal{S}_t$ contains the links that are scheduled in time slot t.

$t \leftarrow 1, \mathcal{S}_t \leftarrow$ NULL, $\mathcal{M} \leftarrow$ NULL;
while $\mathcal{L}$ *is not NULL* **do**
  while $\mathcal{L} \backslash \mathcal{M}$ *is not NULL* **do**
    Find the shortest link l_m in $\mathcal{L} \backslash \mathcal{M}$, $\mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{l_m\}$;
    $\mathcal{L} \leftarrow \mathcal{L} \backslash$ {links that are incident with $x(l)$};
    for *each link l in* $\mathcal{L} \backslash \mathcal{M}$ **do**
      if *l violates the interference checking rules by calling* ***Algorithm 4*** **then**
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{l\}$;
      else
        continue;
      end
    end
  end
  $t \leftarrow t + 1, \mathcal{M} \leftarrow$ NULL;
end

Algorithm 4: Interference Checking for SFAS

Input: A schedule $\mathcal{S}$, a link l, the shortest link l_m.
Output: TRUE if l violates the interference checking rules for the scheduled links in $\mathcal{S}$, FALSE otherwise.

1 $\lambda \leftarrow \frac{2^{\alpha-2}-1}{72\cdot\beta\cdot 2^{\alpha}}$;
2 **if** $RI(l, l_m) \geq \lambda c(l_m)$ **then**
3   **return** TRUE;
4 **else**
5   **for** *each link* l' *in* $ALS(l, \mathcal{S}) \cup \{l\}$ **do**
6   **if** $RI(l', l) < \frac{3\beta}{2+3\beta} \cdot RI(\mathcal{S} \backslash ALS(l', l, \mathcal{S}), l) + \frac{3c(l)-3\beta}{2+3\beta}$ **then**
7   **return** TRUE;
8   **else**
9   **continue**;
10   **end**
11   **end**
12 **end**
13 **return** FALSE;

relative interference from the set of scheduled links. To be more specific, let $\mathcal{S}$ be the set of scheduled links and l be a link waiting for scheduling. If the relative interference of any link $l' \in ALS(l, \mathcal{S}) \cup \{l\}$ on link l is below a certain value as defined in Line 6 of Algorithm 4, then we will consider that link l is unable to be scheduled currently. All the links failing in passing the interference checking process form a temporary set $\mathcal{M}$. We will prove in the next section that this kind of interference checking makes sure that whenever we pick the shortest link among the set $\mathcal{L}\backslash\mathcal{M}$ of remaining links for scheduling, the schedule is certainly successful with SIC, meeting the decoding condition presented in Proposition 2. During each time slot, we will repeat the above steps until all links are either scheduled in or deleted from the current time slot. The whole process is repeated until all the deleted links are scheduled successfully.

5 ALGORITHM ANALYSIS

In this section, we present theoretical analysis about the correctness and efficiency of F2AS and SFAS. We will show that both algorithms produce interference-free schedules and can be fulfield in polynomial time. Here a schedule is said to be interference-free if the transmissions of the active links in each time slot are decoded successfully by the intended receivers with the help of SIC despite of the existence of interference.

Theorem 2. *The schedule of F2AS is interference-free with SIC.*

Proof. In order to prove that the schedule is interference-free, we need to show the transmissions of all the links added to the current time slot are successful. In F2AS, there are two cases where a link is added to $\mathcal{S}_t$. The first is shown in Line 3 of Algorithm 1, where we randomly pick a link l and schedule it. Since it is the only link in $\mathcal{S}_t$, its transmission is surely successful. The second is shown in Line 9 of Algorithm 1, where we schedule a link l' that is confirmed by Algorithm 2 in terms of its feasibility with links in $\mathcal{S}_t$. Since Algorithm 2 is indeed another way of expressing the decoding condition stated in Proposition 2, the transmission of l' is also successful. Therefore, the transmissions of all the links in $\mathcal{S}_t$ are successful, indicating that the schedule of F2AS is interference-free. This finishes the proof.

Theorem 3. *The schedule of SFAS is interference-free with SIC.*

Proof. As before, we need to verify the success of the transmission of any link l added to $\mathcal{S}_t$. Let $\mathcal{P}$ be the set of links added to $\mathcal{S}_t$ before l (i.e. the links shorter than l) and $\mathcal{Q}$ be the set of links added to $\mathcal{S}_t$ after l (i.e. the links longer than l), we have $\mathcal{S}_t = \mathcal{P} \cup \{l\} \cup \mathcal{Q}$. For each scheduled link l in $\mathcal{S}_t$, we can achieve the following two facts easily based on Line 2 and Line 6 in Algorithm 4, respectively.

Fact 1: for each link $l' \in \mathcal{Q}$, we have $RI(l', l) < \lambda c(l)$.

Fact 2: for each link $l' \in ALS(l, \mathcal{P}) \cup \{l\}$, we have $RI(l', l) \geq \frac{3\beta}{2+3\beta} \cdot RI(\mathcal{P} \backslash ALS(l', l, \mathcal{P}), l) + \frac{3c(l)-3\beta}{2+3\beta}$.

We now try to bound the relative interference from all the links in $\mathcal{Q}$ on link l.

Lemma 1. *The relative interference from all links in $\mathcal{Q}$ on l is below $\frac{c(l)}{3\beta}$, i.e. $RI(\mathcal{Q}, l) \leq \frac{c(l)}{3\beta}$.*

The proof of Lemma 1 can be found in Appendix B. According to Fact 1, for each link $l' \in \mathcal{Q}$, we have $\frac{d(l',l)}{d(l)} > (\frac{1}{\lambda})^{\frac{1}{\alpha}}$. Observe that $\lambda > 1$ and $2 < \alpha \leq 6$, so we have $d(l', l) > d(l)$. Consequently, $\mathcal{Q}$ doest not contain any associated link of l, which means $ALS(l, \mathcal{S}_t) \cap \mathcal{Q} =$ NULL and $ALS(l, \mathcal{S}_t) = ALS(l, \mathcal{P})$. Likewise, we have $ALS(l', l, \mathcal{S}_t) \cap \mathcal{Q}$=NULL and $ALS(l', l, \mathcal{S}_t) = ALS(l', l, \mathcal{P})$. Let l' be any link in $ALS(l, \mathcal{S}_t)$. According to Fact 2, we have

$$\begin{aligned}(2+3\beta)RI(l',l) &\geq 3\beta\cdot RI(\mathcal{P}\backslash ALS(l',l,\mathcal{P}),l)+3c(l)-3\beta\\ &= 3\beta\cdot RI(\mathcal{S}_t\backslash(\mathcal{Q}\cup\{l\})\backslash ALS(l',l,\mathcal{S}_t),l)+3c(l)-3\beta\\ &= 3\beta\cdot[RI(\mathcal{S}_t\backslash ALS(l',l,\mathcal{S}_t),l)-RI(\mathcal{Q},l)]+3c(l)-3\beta\\ &\geq 3\beta\cdot RI(\mathcal{S}_t\backslash ALS(l',l,\mathcal{S}_t),l)+3c(l)-3\beta-c(l)\\ &> 3\beta\cdot RI(\mathcal{S}_t\backslash ALS(l',l,\mathcal{S}_t),l)+3c(l)-3\beta-RI(l',l)\end{aligned} \tag{5}$$

The last inequality holds since $RI(l',l)=c(l)\cdot\frac{d^\alpha(l)}{d^\alpha(l',l)}>c(l)$. Thus $RI(l',l)>\frac{\beta}{\beta+1}\cdot RI(\mathcal{S}_t\backslash ALS(l',l,\mathcal{S}_t),l)+\frac{c(l)-\beta}{\beta+1}$, which satisfies the decoding condition stated in Proposition 2. As a result, the transmission of link l is successful with SIC. This finishes the proof.

Theorem 4. *The computational complexity of F2AS and SFAS are both in polynomial time.*

Proof. Firstly, we analyze the computational complexity of F2AS. To check whether a link l does violate the decoding conditions of the links in the scheduled set $\mathcal{S}$, it takes Algorithm 2 at most $|\mathcal{S}|\cdot|ALS(l,\mathcal{S})|\leq|\mathcal{S}|^2$ running time. Furthermore, the while-loop and for-loop in Algorithm 1 both take $O(|L|)$ running time in the worst case. Therefore, for a set L of links, F2AS needs at most $O(|L|^2\cdot|\mathcal{S}|^2)\leq O(|L|^4)$ running time. Thus the complexity of F2AS is $O(|L|^4)$ in the worst case, which is polynomial time.

We then analyze the complexity of SFAS. Similarly, we can figure out that it takes at most $O(|\mathcal{S}|)$ running time for Algorithm 4 to finish the interference checking process for SFAS. In addition, the three loops (two while-loops and one for-loop) in Algorithm 3 all take $O(|L|)$ running time in the worst case. Therefore, the complexity of SFAS can be computed as $O(|L|^3\cdot|\mathcal{S}|)\leq O(|L|^4)$, which is also polynomial time.

6 SIMULATION

6.1 setup

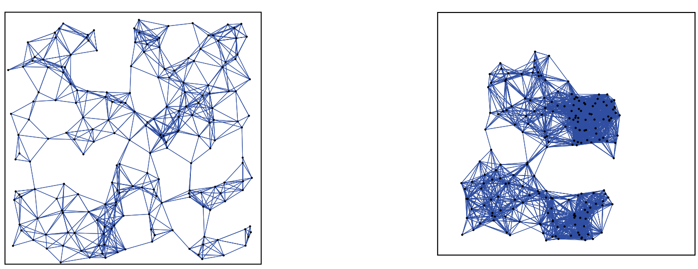
In this section, we evaluate the practical performance of F2AS and SFAS using simulations. We compare both algorithms with two state-of-the-art scheduling algorithms proposed by Xu *et al.* in [31] and Li *et al.* in [18], respectively, which are both designed for MLAS under the physical interference model without SIC. We first briefly introduce these two algorithms.

1. *Xu et al.'s*: First, the algorithm constructs a connected dominator set (CDS) based tree as the routing backbone of data aggregation. Then,

it schedules the nodes on the backbone layer by layer. At each time slot, the algorithm tries to schedule those nodes that are separated far away enough from each other to make sure all the transmissions be interference-free.

2. *Li et al.'s*: First, the algorithm divides all possible links into several categories according to their lengths. Then, the algorithm performs in an iterative fashion and at each iteration, a local leader is selected to collect data from all nodes within a certain distance from it (corresponding to one category of links). Similarly, only links far from each other enough are allowed to be scheduled simultaneously.

For a set of n nodes deployed in a square region of $h \times h$, we consider two typical network topologies, namely *uniform network* and *clustered network*. The former one, where nodes are uniformly and randomly distributed in the plane, is widely used in the research of WSNs for its convenience of mathematical analysis. The latter one, where the centers of n_C clusters are uniformly and randomly located in the plane and in each cluster nodes are uniformly and randomly distributed within a disk of radius r_C at the center, is in fact more suited for practical WSNs for its heterogeneous density distribution [7]. Figure 2 shows sample topologies of 200 nodes with both distributions. In the following simulations for clustered topology, we set the cluster radius r_C as the maximum transmission range r and for the case where a node happens to lie within the ranges of two or more clusters, it joins the nearest one for possibly energy efficiency. Since the network topologies may largely impact the performance of our algorithms involving geometric properties, we generate 50 topologies for each simulation and compare different algorithms using average results.

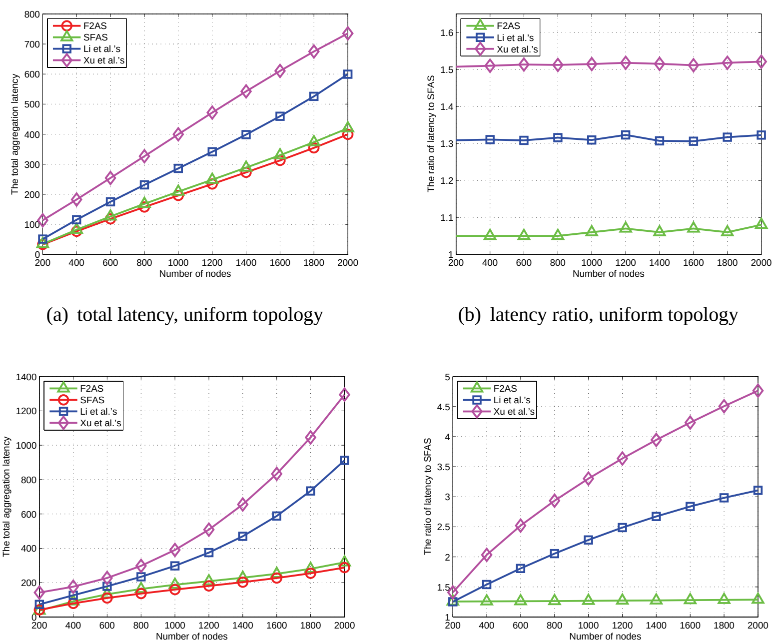


(a) Uniform network (b) Clustered network

FIGURE 2
Sample topologies of 200 nodes with different distributions

6.2 impact of network size

Firstly, we evaluate the impact of the number of nodes on the aggregation latency of all algorithms. We set the parameters α, β, ξ and P of the physical interference model as 2.5, 1.0, 0.1, and 15, respectively, the same with those in [31]. Base on these parameters, the maximum transmission range r can be computed as $r = (\frac{P}{\xi\beta})^{\frac{1}{\alpha}} = 7.4$. We fix h as $100m$ and vary n from 200 to 2000. For clustered topology, the cluster radius r_C is set as 7.4 (equals to r) and the cluster number n_C is chosen as a moderate value of 30. Later we will consider different values of n_C. Figure 3(a)-(b) and (c)-(d) present the simulation results under uniform network and clustered network, respectively. We can see that both F2AS and SFAS have better performance compared with Li *et al.*'s and Xu *et al.*'s under various network size for uniform and clustered topologies. In uniform network, the latency of Li *et al.*'s and Xu *et al.*'s are 30% and 50% longer than that of SFAS, respectively, as shown in Figure 3(b). What's more, in clustered network, the latency of Li *et al.*'s and Xu *et al.*'s can be 3.6 times and 4.7 times as long as that of SFAS, respectively, as shown in Figure 3(d). In other words, the performance improvement



(a) total latency, uniform topology

(b) latency ratio, uniform topology

(c) total latency, clustered topology

(d) latency ratio, clustered topology

FIGURE 3
Aggregation latency with varied number of nodes

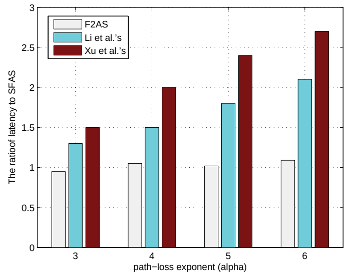
of our algorithms compared with Li *et al.*'s and Xu *et al.*'s is more remarkable in clustered network than that in uniform network. This is due to the fact that in uniform network, the link lengths do not diverse and little benefit from SIC can be gained, while in clustered network, the link lengths diverse much and as a result, much more benefit from SIC are obtained. In addition, we can see that as the density of nodes increases, the relative performance of SFAS and F2AS becomes better and better in clustered network as shown in Figure 3(d), which indicates that both algorithms can schedule many clusters in parallel while Li *et al.*'s and Xu *et al.*'s algorithms can not achieve it. While in uniform network, the relative performance of F2AS and SFAS does not change much as the density of nodes increases as shown in Figure 3(b). This is because that although the density of links increases, the diversification of link lengths does not change much, which prevents F2AS and SFAS from gaining more benefit from SIC. For F2AS and SFAS which both benefit from SIC, we can see that they have comparable performance under various network conditions, and the gap between them does not exceed 10%.

6.3 impact of SINR parameters α, β

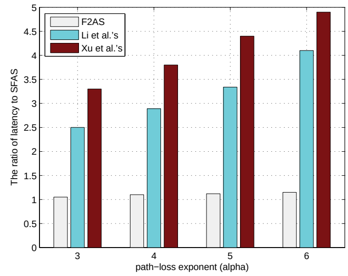
We then evaluate the impact of SINR parameters α and β. We set ξ, P and h as 0.1, 15 and 100, respectively, and consider a moderate network density of 1000 nodes. For clustered topology, we set $n_C = 30$, the same as above and let $r_C = r$, whose value varies when α and/or β are changing. Figure 4(a) and (b) show the impact of α on the aggregation latency of different algorithms. We observe that as α increases, the relative performance of SFAS and F2AS becomes increasingly much better than that of Li *et al.*'s and Xu *et al.*'s algorithms in both uniform and clustered networks. This is due to the fact that the larger the value of α is, the more different the received signal strength of the links will be. Consequently, more benefit from SIC can F2AS and SFAS gain. Figure 4(c) and (d) show the impact of SINR threshold β. It can be seen that the value of β does not influence the performance of any algorithm much, which is expected, given that β is just a ratio.

6.4 impact of cluster number n_C

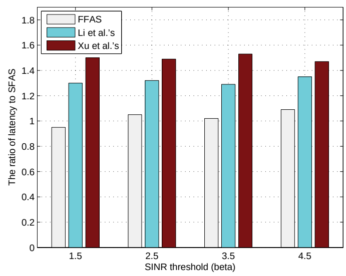
From previous simulations we can observe that our algorithms perform much better in clustered network where more transmission opportunities can be exploited from SIC. However, we all set the cluster number n_C as a moderate value of 30 empirically. Since the value of n_C affects the geometric feature of clustered network and hence may further influences the performance of different scheduling algorithms, we will investigate the impact of n_C on aggregation latency in this section. Considering a network of 1000 nodes and assuming that α, β, ξ, P are set as 2.5, 1.0, 0.1, and 15, respectively, we plot the latency curves of all algorithms with varied n_C in Figure 5. Note that n_C



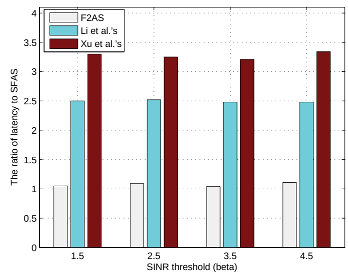
(a) α, uniform topology



(b) α, clustered topology

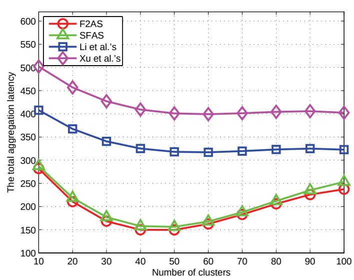


(c) β, uniform topology

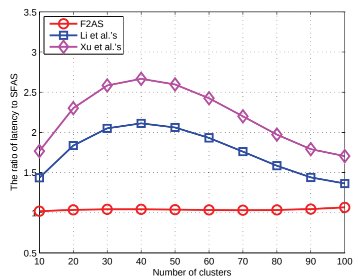


(d) β, clustered topology

FIGURE 4
Aggregation latency ratio with varied SINR parameters α and β



(a) n_C, clustered topology



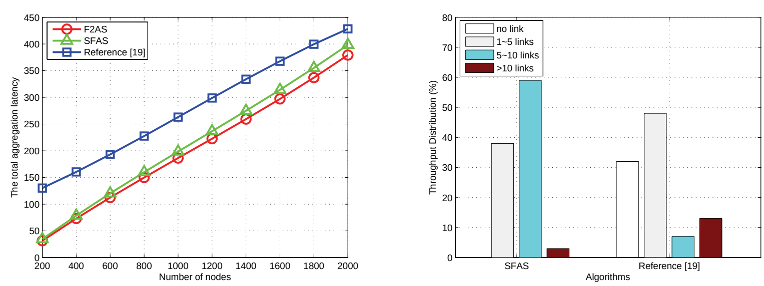
(b) n_C, clustered topology

FIGURE 5
Aggregation latency with varied n_C in clustered network

can not be too small as all the clusters located randomly in the plane should form a connected graph. From Figure 5(a), we can see that all the algorithms perform worst when there is only 10 clusters within the network, since the number of clusters is so small that lots of links are crowded, preventing any algorithm from scheduling multiple links simultaneously. As n_C becomes larger, the latency performance of all algorithm becomes better while F2AS and SFAS achieve much shorter schedules than those of Li *et al.*'s and Xu *et al.*'s, owing to the new transmission opportunities offered by SIC. Specifically, the latency of Li *et al.*'s and Xu *et al.*'s are 2.6 times and 2.1 times that of SFAS respectively, as shown in Figure 5(b). We can also observe that as n_C becomes very large (exceeds 90), the gap between SFAS with Li *et al.*'s or Xu *et al.*'s algorithms becomes smaller. In other words, little relative performance improvement can be gained by SFAS, possibly due to the fact that in this case, the clustered network tends to be a uniform one, where link lengths diverse little and at a result, not that much benefit can be obtained from SIC, as we have discussed in Section 6.2.

6.5 Comparison with Reference [19]

Finally, we conduct simulations to compare F2AS and SFAS with the recently published work [19] which also targets data aggregation with SIC enabled as we have discussed in Section 2.3. We generate a network topology where the distance between any pair of nodes are longer than $1m$ as required by [19]. The parameters α, β, ξ, h and P are set as 2.5, 1, 0.1, 100 and 15, respectively. For F2AS and SFAS, P is the uniformly used power by all nodes while for [19], P is the bounded maximum power available with power control. The latency curves of all algorithms with varied network size n from 200 to 2000 are shown in Figure 6(a). We can observe that the latency of F2AS and SFAS are both shorter than that of Reference [19]'s under various values of n. This may be explained for that in [19], the benefit of SIC is only exploited at the first step of data aggregation where local headers aggregate data from neighbors, while at the following steps where data are delivered from local headers to backbone headers and further to the sink, no SIC opportunity is exploited at



(a) aggregation latency (b) throughput distribution, $n = 1000$

FIGURE 6
Performance comparison between SFAS and Reference [19]

all, as can be found in Algorithm 1 of [19]. In our heuristics, we fully exploit the benefit from SIC at each step and during each time slot, thereby achieving much better latency performance. We further explore the detail of schedules produced by our heuristics and [19]. Taking SFAS for example and considering a network of 1000 nodes, we plot the throughput distributions of SFAS and [19] in Figure 6(b). Here the *throughput distribution* is the distribution of the number of concurrent active links over the whole schedule period. We can observe that [19] provides more chances for 10 or more links to be scheduled concurrently than SFAS. Nevertheless, in the most cases, SFAS can schedule at many as 5$\sim$10 links simultaneously while [19] can only schedule 1$\sim$5 links in general. In addition, we can also observe that there exist many idle time slots in which no link is scheduled at all by [19] while SFAS can always schedule at least one link in each time slot. To be more fair, though the work in [19] does not have that nice latency performance as ours, it has its unique merit of easy implementation. Therefore, it is necessary to develop efficient scheduling algorithms for data aggregation which not only fully exploit the benefit of SIC but also are amendable to distributed implementation, which we take as part of our future work.

In a word, the simulations above show that SIC does increase transmission opportunities and improve the latency of data aggregation operation significantly, provided that it is carefully identified and fully exploited. What's more, the network topology and SINR parameters do have considerable effect on the extent to which the scheduling algorithms can benefit from SIC.

7 CONCLUSION AND FUTURE WORK

In this paper, we study the minimum-latency aggregation scheduling problem under the physical interference model with successive interference cancellation (termed MLAS-SIC). We formulate the MLAS-SIC problem and propose two heuristic scheduling algorithms, namely F2AS and SFAS by fully exploiting the new transmission opportunities offered by SIC. Theoretical analysis indicates that both algorithms are correct and can achieve polynomial computational complexity. Simulation results show that our algorithms outperform state-of-the-art scheduling algorithms designed for MLAS under the physical interference model without SIC. Our work reveals that when SIC is enabled at the physical layer, the performance of carefully designed scheduling algorithms can be improved significantly in WSNs.

Several interesting questions are left for further research. The first one is to derive an efficient algorithm for MLAS-SIC with guaranteed approximation performance with respect to the optimal solution. The second one is to develop distributed implementations for our algorithms, which will be more suitable for large-scale and time-varying WSNs. The third one is to apply the design idea raised in this paper to other important operations in wireless networks such as gossiping, broadcast [15] and data collection [10].

REFERENCES

[1] Min Kyung An, Nhat X. Lam, Dung T. Huynh, and Trac N. Nguyen. (2012). Minimum latency data aggregation in the physical interference model. *Computer Communications*, 35(18):2175–2186.

[2] Xujin Chen, Xiaodong Hu, and Jianming Zhu. (2005). Minimum data aggregation time problem in wireless sensor networks. In *Mobile Ad-hoc and Sensor Networks*, pages 133–142. Springer.

[3] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. (2001). *Introduction to algorithms*. MIT press.

[4] Hongwei Du, Zhao Zhang, Weili Wu, Lidong Wu, and Kai Xing. (2012). Constant-approximation for optimal data aggregation with physical interference. *Journal of Global Optimization*, pages 1–14.

[5] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*, 14(2):70–87.

[6] Majid Ghaderi and Mohsen Mollanoori. (2013). Uplink scheduling in wireless networks with successive interference cancellation. *Mobile Computing, IEEE Transactions on*, to appear.

[7] Olga Goussevskaia, Roger Wattenhofer, Magnús M Halldórsson, and Emo Welzl. (2009). Capacity of arbitrary wireless networks. In *INFOCOM 2009, IEEE*, pages 1872–1880. IEEE.

[8] Piyush Gupta and Panganmala R Kumar. (2000). The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404.

[9] Daniel Halperin, Thomas Anderson, and David Wetherall. (2008). Taking the sting out of carrier sense: interference cancellation for wireless lans. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 339–350. ACM.

[10] Liang He, Jianping Pan, and Jingdong Xu. (2013). Data Collection Latency in Wireless Sensor Networks with Multiple Mobile Elements. *AD HOC & SENSOR WIRELESS NETWORKS*, 18(1-2):109–129.

[11] Nathaniel Hobbs, Yuexuan Wang, Qiang-Sheng Hua, Dongxiao Yu, and Francis CM Lau. (2012). Deterministic distributed data aggregation under the sinr model. In *Theory and Applications of Models of Computation*, pages 385–399. Springer.

[12] SC-H Huang, Peng-Jun Wan, Chinh T Vu, Yingshu Li, and Frances Yao. (2007). Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 366–372. IEEE.

[13] Canming Jiang, Yi Shi, Y Thomas Hou, Wenjing Lou, Sastry Kompella, and Scott F Midkiff. (2012). Squeezing the most out of interference: An optimization framework for joint interference exploitation and avoidance. In *INFOCOM, 2012 Proceedings IEEE*, pages 424–432. IEEE.

[14] Xianlong Jiao, Wei Lou, Junchao Ma, Jiannong Cao, Xiaodong Wang, and Xingming Zhou. (2012). Minimum latency broadcast scheduling in duty-cycled multihop wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(1):110–117.

[15] Xianlong Jiao, Xiaodong Wang, Wei Lou, Jiannong Cao, Xiao Xia, Xingming Zhou, and Geming Xia. (2013). On Minimizing Interference-Free Broadcast Latency in Duty-Cycled Wireless Sensor Networks. *AD HOC & SENSOR WIRELESS NETWORKS*, 18(3-4, SI):293–309.

[16] Nabila Labraoui, Mourad Guerroui, Makhlouf Aliouat, and Tanveer Zia. (2011). Data Aggregation Security Challenge in Wireless Sensor Networks: A Survey. *AD HOC & SENSOR WIRELESS NETWORKS*, 12(3-4):295–324.

[17] Deying Li, Qinghua Zhu, Hongwei Du, Weili Wu, Hong Chen, and Wenping Chen. (2011). Conflict-free many-to-one data aggregation scheduling in multi-channel multi-hop wireless sensor networks. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE.

[18] Hongxing Li, Chuan Wu, Qiang-Sheng Hua, and Francis Lau. (2011). Latency-minimizing data aggregation in wireless sensor networks under physical interference model. *Ad Hoc Networks*, 23(2):123–135.

[19] Hongxing Li, Chuan Wu, Dongxiao Yu, Q Hua, and F Lau. (2013). Aggregation latency-energy tradeoff in wireless sensor networks with successive interference cancellation. *Parallel and Distributed Systems, IEEE Transactions on*, to appear.

[20] Yingshu Li, Longjiang Guo, and Sushil K Prasad. (2010). An energy-efficient distributed algorithm for minimum-latency aggregation scheduling in wireless sensor networks. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 827–836. IEEE.

[21] Shaohe Lv, Weihua Zhuang, Xiaodong Wang, and Xingming Zhou. (2011). Scheduling in wireless ad hoc networks with successive interference cancellation. In *INFOCOM, 2011 Proceedings IEEE*, pages 1287–1295. IEEE.

[22] Guanhong Pei and VS Anil Kumar. (2013). Distributed approximation algorithms for maximum link scheduling and local broadcasting in the physical interference model. In *INFOCOM, 2013 Proceedings IEEE*. IEEE.

[23] Souvik Sen, Naveen Santhapuri, R Choudhury, and Srihari Nelakuditi. (2013). Successive interference cancellation: Carving out mac layer opportunities. *Mobile Computing, IEEE Transactions on*, to appear.

[24] David Tse and Pramod Viswanath. (2005). *Fundamentals of wireless communication*. Cambridge university press.

[25] Peng-Jun Wan, Dechang Chen, Guojun Dai, Zhu Wang, and Frances Yao. (2012). Maximizing capacity with power control under physical interference model in duplex mode. In *INFOCOM, 2012 Proceedings IEEE*, pages 415–423. IEEE.

[26] Peng-Jun Wan, Scott C-H Huang, Lixin Wang, Zhiyuan Wan, and Xiaohua Jia. (2009). Minimum-latency aggregation scheduling in multihop wireless networks. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 185–194. ACM.

[27] Guanyu Wang, Qiang-Sheng Hua, and Yuexuan Wang. (2012). Minimum latency aggregation scheduling for arbitrary tree topologies under the sinr model. In *Ad-hoc, Mobile, and Wireless Networks*, pages 139–152. Springer.

[28] Xin Wang and JJ Garcia-Luna-Aceves. (2009). Embracing interference in ad hoc networks using joint routing and scheduling with multiple packet reception. *Ad Hoc Networks*, 7(2):460–471.

[29] Steven P Weber, Jeffrey G Andrews, Xiangying Yang, and Gustavo De Veciana. (2007). Transmission capacity of wireless ad hoc networks with successive interference cancellation. *Information Theory, IEEE Transactions on*, 53(8):2799–2814.

[30] XiaoHua Xu, Xiang-Yang Li, XuFei Mao, Shaojie Tang, and Shiguang Wang. (2011). A delay-efficient algorithm for data aggregation in multihop wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(1):163–175.

[31] Xiaohua Xu, Xiang-Yang Li, and Peng-Jun Wan. (2013). Efficient aggregation scheduling in multihop wireless sensor networks with sinr constraints. *Mobile Computing, IEEE Transactions on*, to appear.

[32] Bo Yu, Jianzhong Li, and Yingshu Li. (2009). Distributed data aggregation scheduling in wireless sensor networks. In *INFOCOM 2009, IEEE*, pages 2159–2167. IEEE.

A PROOF OF THEOREM 1

Let l be a link to be scheduled. With SIC, we must first check whether any link l' ($P(l', l) > P(l)$) obeys the inequality (2) in Section 3.1 or not. Let the SINR threshold β in (2) be infinite, then l' is surely unable to be decoded at $r(l)$. Thus, only when the signal from $x(l)$ is the largest at $r(l)$ and $P(l)$ obeys the inequality (3) in Section 3.1 can the transmission of l be successful. At such circumstances, the problem reduces to the one without SIC, which has been proved to be NP-hard in [1]. As a result, our problem, as a more general case, is also NP-hard. This finishes the proof.

B PROOF OF LEMMA 1

According to Fact 1, for each link $l' \in \mathcal{Q}$, we have $RI(l', l) = c(l) \cdot \frac{d^\alpha(l)}{d^\alpha(l',l)} < \lambda \cdot c(l)$, thus $d(l', l) > \sqrt[\alpha]{\frac{1}{\lambda}} d(l)$. Let l_i, l_j be any two links in $\mathcal{Q}$, we have $d(x(l_i), x(l_j)) \geq d(x(l_i), r(l_j)) - d(l_j) > (\sqrt[\alpha]{\frac{1}{\lambda}} - 1)d(l)$. In other words, the distance between any two transmitters in $\mathcal{Q}$ is longer than $(\sqrt[\alpha]{\frac{1}{\lambda}} - 1)d(l)$. The following lemma has been verified in [7].

Lemma 2. [7] *Let $\mathcal{N}$ be a set of nodes and Λ_κ be the κ-th ring in the plane composed of two concentric circles with radius $\kappa \cdot \theta$ and $(\kappa + 1) \cdot \theta$ ($\kappa = 0, 1, 2 \cdots$), respectively. Provided that the distance between any two nodes in $\mathcal{N}$ is no shorter than θ', then the number of nodes in $\mathcal{N}$ that are located within Λ_κ can be bounded as* $8(2\kappa + 1)(\frac{\theta}{\theta'})^2$.

Let $r(l)$ be the common center of the rings and $\theta = \sqrt[\alpha]{\frac{1}{\lambda}} d(l)$, $\theta' = (\sqrt[\alpha]{\frac{1}{\lambda}} - 1)d(l)$. Based on Lemma 2, we can bound the total relative interference from all nodes located in Λ_κ as below:

$$\begin{aligned} RI(\Lambda_\kappa, l) &= \sum_{l_i \in \Lambda_\kappa} RI(l_i, l) \\ &\leq 8(2\kappa + 1) \cdot (\frac{\theta}{\theta'})^2 \cdot c(l) \cdot \frac{d^\alpha(l)}{(\kappa \cdot \theta)^\alpha} \\ &= 8 \cdot \frac{2\kappa + 1}{\kappa} \cdot (\frac{\theta}{\theta'})^2 \cdot (\frac{d(l)}{\theta})^\alpha \cdot c(l) \cdot \kappa^{1-\alpha} \\ &\leq 96\lambda \cdot c(l) \cdot \kappa^{1-\alpha} \end{aligned} \tag{6}$$

The last inequality holds since $2\kappa + 1 \leq 3\kappa$, $\theta \leq 2\theta'$ and $d(l) = \sqrt[n]{\lambda} \cdot \theta$. Summing up the relative interference from all rings Λ_κ ($\kappa = 1, 2, 3, ..., \kappa \neq 0$ since no concurrent transmitter exists in Λ_0), we have

$$\begin{aligned} RI(\mathcal{Q}, l) &= \sum_{\kappa=1}^{\infty} RI(\Lambda_\kappa, l) \\ &\leq 96\lambda \cdot c(l) \cdot \sum_{\kappa=1}^{\infty} \kappa^{1-\alpha} \\ &\leq 96\lambda \cdot c(l) \cdot \frac{2^\alpha}{2^\alpha - 4} \\ &= \frac{c(l)}{3\beta} \end{aligned} \tag{7}$$

This finishes the proof.