Improved Minimum Latency Aggregation Scheduling in Wireless Sensor Networks under the SINR Model

Zhaoquan Gu
Institute for Interdisciplinary Information Sciences, Tsinghua University, 100084, Beijing, P.R. China
E-mail: demin456@gmail.com

Guanyu Wang
Institute for Interdisciplinary Information Sciences, Tsinghua University, 100084, Beijing, P.R. China
E-mail: wgiveny@gmail.com

Qiang-Sheng Hua
Institute for Interdisciplinary Information Sciences, Tsinghua University, 100084, Beijing, P.R. China
E-mail: qshua@mail.tsinghua.edu.cn

Yuexuan Wang*
Institute for Interdisciplinary Information Sciences, Tsinghua University, 100084, Beijing, P.R. China
E-mail: wangyuexuan@tsinghua.edu.cn
*Corresponding author

Abstract: Wireless sensor networks are attracting much attention from the world and Minimum Latency Aggregation Scheduling (MLAS) has become one of the most significant fundamental problems in wireless sensor networks. However, there are few results about efficient data aggregation algorithms under the Signal-to-Interference-plus-Noise-Ratio (SINR) model. In this paper, we propose a centralized algorithm to aggregate data from all sources in $O(\log^2 n)$ time slots where $n$ is the total number of nodes. To the best of our knowledge, this is the current best result for the problem. This algorithm uses round scheduling, topology construction and non-linear power assignment as the main techniques. We give a detailed proof of correctness, also an aggregation latency analysis of the algorithm as well as the parameter constraints to achieve our result.

Keywords: wireless sensor network, data aggregation, minimum latency, algorithm

1 Introduction

Nowadays, the wireless sensor networks have been attracting vast attention for their wide usage in many industrial and consumer applications, such as environmental monitoring, machine health monitoring and control, etc. One of the significant fundamental problems in the wireless sensor networks (and the wireless networks in general) is the efficient method to collect data from individual nodes. More precisely, given a set of sensor nodes which have their own sensing data, arbitrarily distributed in a metric space, how efficiently can these nodes transfer their data to the sink node? This question can be formulated as: what is the minimum number of time slots (we divide the time into unit slots) that can be used to schedule all the aggregation transmissions without collision under the SINR model. This is so called Minimum − Latency Aggregation Scheduling (MLAS) problem (Chen, Hu, and Zhu, 2005; Huang et al., 2007; Wan et al., 2009; Yu, Li, and Li, 2009).

Why is this problem so important? In any real wireless sensor network application, each node in the network has to send its own sensed data to the sink node frequently. Actually, many query applications are handling such a MLAS problem, for instance, querying the max or min temperature in a large area. So, how the
sink node can gather all the data efficiently in a timely fashion is an interesting and practical problem.

In studying wireless sensor networks, communication model is important when it comes to algorithm design. There are two common models for wireless communication: the Protocol Model and the Physical Model (or Signal to Interference plus Noise Ratio, SINR model) (Gupta and Kumar, 2000). In many wireless sensor network research papers, multi-hop wireless networks have been modeled as graphs. All nodes of this communication graph represent the physical devices, two nodes being connected if and only if the respective devices are within mutual transmission range. In the graph-based model, a node is assumed to receive a message successfully if and only if no other node in physical proximity transmits at the same time. It is foreseeable that in graph theory, interference-free concurrent transmissions just boil down to solving variants of coloring or independent set problems.

Compared with the tremendously simplified graph-theoretic model, the SINR model is a more accurate description of reality. The advantage and robustness of the SINR model have been analyzed (Thomas and Roger, 2006; Magnus and Roger, 2010). In this paper, we adopt the physical model to study the data gathering problem.

### 1.1 Related work

Even though the MLAS problem is a fundamental problem of wireless sensor networks, there has been little work done under the physical model. Many related works focusing on solving this problem are under the protocol model (Chen, Hu, and Zhu, 2005; Huang et al., 2007; Wan et al., 2009; Xu et al., 2009; Yu, Li, and Li, 2009). (Chen, Hu, and Zhu, 2005) provides an algorithm for MLAS within $O((\Delta - 1)R)$ time slots, we also call it aggregation latency, where $\Delta$ is the maximum node degree and $R$ is the network radius defined by hop count. The NP-hardness of this problem is also proved in this paper. Under the protocol model, the best results (Huang et al., 2007; Wan et al., 2009; Xu et al., 2009; Yu, Li, and Li, 2009) show that aggregation latency can be bounded by $O(\Delta + R)$. Some other factors have also been researched such as energy control in (Hua and Lau, 2006; Moh, Kim and Moh, 2006) where Moh et al. present a distributed power scheduling for data aggregation, routing protocol design in (Jia, Zhao and Ma, 2008) which improves the life span of the network. In addition, maximizing the lifetime with data aggregation in wireless sensor networks have also been discussed in (Stanford and Tongngam, 2009; Li, Zhu and Chen, 2011; Zou, Nikolaidis and Harms, 2008) and efficient aggregation tree constructed in (Cheng and Yin, 2008; Chiang and Byrd, 2009; Hua and Lau, 2010) can reduce redundant data which improves the aggregation latency. (Cam, 2007) also gives a view about coding method in data aggregation and (Kafatzoglou and Papavassiliou, 2011; Solis and Obrazcza, 2006) explore the trade-off for data collection when in-network aggregation is introduced. Some distributed algorithms for local broadcast are also discussed in (Hua et al., 2011; Yu et al., 2011a; Yu et al., 2011b).

To the best of our knowledge, there are few papers (Li et al., 2009; Li et al., 2010; Nathaniel et al., 2012) solving the MLAS problem under the SINR model, in which both centralized and distributed algorithms are given.

The first solution in (Li et al., 2009) proposes a distributed algorithm using constant power assignment. This algorithm can produce a feasible scheduling for aggregation transmissions with latency at most $O(\Delta + R)$. It is obvious that the efficiency of this algorithm depends on the networks’ topologies, which may result in $O(n)$ latency in the worst case. Moreover, this paper only takes constant power assignment into account. However, the discussion in (Thomas and Roger, 2006) had already shown uniform power assignment will not work efficiently in some worst scenarios of the scheduling problem in wireless networks.

The best MLAS solution before our result is given in (Li et al., 2010) which develops both distributed and centralized algorithm. By first aggregating data from sensor nodes in each divided smaller area with shorter transmission links, then repeating the same process for larger areas and longer links until the entire network is covered by the largest area, the distributed algorithm achieves a latency bounded by $O(K)$, where $K$ is the logarithm of the ratio between the lengths of the longest and shortest links in the network, which can be $O(n)$ ($n$ is the total number of nodes) in the worst case. The centralized algorithm can finish data aggregation in $O(\log^2 n)$ time slots based on a result (Thomas, 2007) from the wireless network capacity problem.

### 1.2 Our contribution

The main result of this paper is an improved centralized algorithm solving the MLAS with an aggregation latency bounded of $O(\log^2 n)$. Our latency bound removes a $O(\log n)$ factor from (Li et al. 2010), which is the best result before ours. We adopt several useful and common techniques, like dividing links into different length groups and non-oblivious power assignments that are used in some related papers (Alexander, Thomas and Berthold, 2009; Dariusz and Mariusz, 2010; Thomas, Roger and Aaron, 2006; Thomas and Roger, 2006; Thomas, 2007). In fact, directly applying the subroutine (Algorithm 4) of our algorithm after constructing the nearest neighbor tree, we can solve the Connectivity Problem in (Thomas, 2006) within $O(\log n)$ time slots, which is also the best known result.

In this paper, we also provide a detailed analysis about the constraints of the parameters involved, which can be helpful in real implementation.

The rest of the paper is structured as follows: we start by introducing the considered wireless models and notations in Section 2. Then we propose our efficient...
improved MLAS algorithm in Section 3. The correctness of this algorithm is given in Section 4. The aggregation latency of our algorithm is analyzed in Section 5. Also we discuss the parameter constrains in Section 6. Finally, we conclude in Section 7.

2 Notation and Model

For a given a set of nodes \( V = \{v_1, v_2, \ldots, v_n\} \), the Euclidean distance between any two node \( v_i, v_j \) is denoted by \( d(v_i, v_j) \). Each link \( l_i = (v_i, v_j) \) represents a communication request from sender \( v_i \) to receiver \( v_j \). All the nodes are distributed in the Euclidean plane, note that all nodes can be both sender and receiver, but only in different time slots (i.e. no node can send and receive simultaneously). The length of link \( l_{ij} \) is denoted by \( d_{ij} = d(v_i, v_j) \). And the distance from link \( l_{gh} \) to link \( l_{ij} \) is the distance from \( l_{gh} \)’s sender to \( l_{ij} \)’s receiver, denoted by \( d_{gj} = d(v_i, v_j) \).

The signal power \( P_t(v_j) \) or simply \( P_t(j) \), received at \( v_j \) from sender \( v_i \) depends on the transmission power \( P_t(j) \) of sender \( v_i \) and the distance \( d_{ij} \) between nodes \( v_i \) and \( v_j \). This is the path loss radio propagation model for the reception of signals, where the signal strength is assumed to fall off with \( d^{-\alpha} \) (\( \alpha > 2 \) denotes the path-loss exponent), i.e. \( P_t(j) = P_t(j)/d_{ij}^\alpha \). Every sender \( v_k \) (with corresponding receiver \( v_k \)) that sends concurrently with \( v_i \) causes an interference \( I_t(j) = P_t(j) = P_{gh}/d_{gj}^\alpha \) at receiver \( v_j \) of link \( l_{ij} \). The notation \( I_t(j) \) is used in order to emphasize that this is interference, not a useful signal.

All interferences accumulate. The total interference \( I_t(v_j) \) experienced by receiver \( j \) is given as the sum of all interferences caused by other concurrently sending nodes, i.e. \( I_t(v_j) = \sum_{l_{ij} \neq l_k} I_t(j) \). A receiver \( v_j \) successfully receives a message from its sender \( v_i \) if and only if it suits the precedence constraint (A node cannot send its data to the parent node until it has received all data from the its children nodes) and the following condition:

\[
P_t(j) \geq \sum_{l_{gh} \in S \setminus l_{ij}} I_t(g) + N \geq \beta
\]

where \( N \) is ambient noise, \( \beta \) denotes the minimum SINR(Signal-to-interference-plus-noise-ratio) required for a message to be successfully received, and \( S \) is the set of concurrently transmitting links, i.e., the links that can be scheduled in the same time slot.

3 Improved MLAS Algorithm

In this section, we present the improved MLAS algorithm such that all links constructed can be scheduled in \( O(\log^2 n) \) time slots for any placement of \( n \) nodes in the plane.

Define ActiveNodeSet as the set of nodes that have not finished sending their data and ActiveLinkSet as the set of links that can be chosen to schedule. We use notations ANS and ALS respectively for short. In the algorithm, \( k, b, c_1, a_1 \) are constant parameters that will be discussed in Section 6.

Algorithm 1 Improved MLAS Algorithm

1. \( \text{ANS} := V \setminus \{\text{sinknode}\}, \text{Tree} := \emptyset \)
2. while \( |\text{ANS}| > 1 \) do
3. \( T := \text{Generate Topology} (\text{ANS}); \)
4. \( T^\prime := \text{Choose Link Set} (T); \)
5. \( \text{Schedule}(T^\prime); \)
6. \( \text{Tree} := \text{Tree} \cup T^\prime; \)
7. for each \( l_{ij} \in T^\prime \) do
8. \( \text{ANS} := \text{ANS} \setminus \{v_i\}; \)
9. end for
10. end while

Algorithm 1 plays the main role in scheduling nodes in round; it schedules all nodes until only single one remains. Then finish the communication between this node and the sink node with one more extra time slot. In each round, generate topology of Active Node Set, construct link set \( T^\prime \) in \( T \) and schedule all links in \( T^\prime \). Now we present each phase of the algorithm and discuss the related properties.

Algorithm 2 Generate Topology on Node Set \( V \)

1. \( T := \emptyset; \)
2. while \( |V| > 1 \) do
3. for each \( v_i \in V \) do
4. \( \text{Find} v_j \in V \setminus \{v_i\} \) minimizing \( d(v_i, v_j); \)
5. if \( l_{ij} \notin T \) then
6. \( T := T \cup \{l_{ij}\}; \)
7. end if
8. end for
9. for each \( l_{ij} \in T \) do
10. \( V := V \setminus \{v_i\}; \)
11. end for
12. end while
13. return \( T; \)

Algorithm 2 uses the nearest neighbor tree method to generate the topology, which has been applied in many papers such as (Dariusz and Mariusz, 2010; Thomas and Roger, 2006; Thomas, 2007). Here are some properties result from this topology:

Property 3.1: Consider two links \( l_{ij} \) and \( l_{ji} \), there is at most one link in tree \( T \).

Property 3.2: Consider link \( l_{ij} \in T \), if there exists another node \( v_k \) with \( d_{ik} < d_{ij} \), \( l_{ki} \) must belongs to the tree \( T \).

Algorithm 3 chooses appropriate links, i.e. these which meet the requirement in the algorithm.

Property 3.3: \( \forall l_{ij} \in T^\prime \), there exists no node \( v_k \) s.t \( l_{ki} \in T^\prime \).
Thus the precedence constraint holds.

rounds.

scheduling all links in

Proof: From Line 4 if there exists such a node $v_k$ s.t $l_{k_i} \in T'$, $l_{ij}$ will be deleted from $\text{ALS}$ at Line 11. □

Property 3.4: All links in $T'$ can be concurrently scheduled satisfying the precedence constraint.

Proof: From Line 10, $\forall l_{ij} \in T'$, $v_j$ has no outgoing edge. Thus the precedence constraint holds. □

Property 3.5: Algorithm 1 runs no more than $\lfloor \log n \rfloor$ rounds.

Proof: It is easy to see that at least $\frac{|\text{ANS}| - 1}{2}$ nodes will be deleted in each round from Algorithm 3. So Algorithm 1 will terminate in no more than $\lfloor \log n \rfloor$ rounds. □

Algorithm 4 is an important phase of our algorithm scheduling all links in $T'$. First divide links into subsets according to their $\gamma$ and $\tau$ values, defined in the pre-processing at Algorithm 4. This division method by link length is commonly used in scheduling algorithms such as (Alexander, Thomas and Berthold, 2009; Thomas, Roger and Aaron, 2006; Thomas and Roger, 2006; Thomas, 2007). Now we will give some properties about the scheduling phase:

Algorithm 3 Choose Link Set on Tree $T$
1: $\text{ALS} := T, T' = \emptyset$
2: while $|\text{ALS}| > 0$ do
3: for each $l_{ij} \in \text{ALS}$ do
4: if there is no such node $v_k, l_{k_i} \in \text{ALS}$ then
5: $T' := T' \cup \{l_{ij}\}$
6: end if
7: end for
8: for each $l_{ij} \in T'$ do
9: $\text{ALS} := \text{ALS} \setminus \{l_{ij}\}$
10: if there exists node $v_k, l_{jk} \in \text{ALS}$ then
11: $\text{ALS} := \text{ALS} \setminus \{l_{jk}\}$
12: end if
13: end for
14: end while
15: return $T'$

Algorithm 4 Schedule $T'$
1: $T' := \text{pre-Processing}(T')$
2: for $m = 1$ to $a_1$ do
3: Let $T_m = \{l_{ij} \in T' | \gamma_{ij} = m\}$
4: while not all links in $T_m$ have been scheduled do
5: $L_i := \emptyset$
6: Order all links in $T_m$ by decreasing order of length;
7: if $\text{canSchedule}(l_{ij}, L_i)$ then
8: $L_i := L_i \cup l_{ij}$
9: end if
10: For all $l_{ij} \in L_i$, set the time slot $t(l_{ij}) := t$ and assign the power $P_t(l_{ij}) := k \cdot b^{\gamma_{ij}} \cdot d_{ij}^\tau$
11: $t := t + 1$
12: end while
13: end for

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Example of 7 nodes in the plane. In every round, the bold black links are chosen to be scheduled. At the end of each round, the nodes that have transmitted their message will be deleted from the ANS. Continue until only one node is left.}
\end{figure}

Pre-processing$(T')$
1: $\tau_{cur} := 1; \gamma_{cur} := 1; \text{last} := d_{ij}$ which is the longest link $l_{ij}$ in $T'$
2: for each $l_{ij}$ in decreasing order of the length $d_{ij}$ do
3: if $\frac{\text{last}}{d_{ij}} \geq 2$ then
4: if $\gamma_{cur} < a_1$ then
5: $\gamma_{cur} := \gamma_{cur} + 1$
6: else
7: $\gamma_{cur} := 1; \tau_{cur} := \tau_{cur} + 1$
8: end if
9: last := $d_{ij}$
10: end if
11: end for

\text{canSchedule}$(l_{ij}, L_i)$
1: for each $l_{gh} \in L_i$ do
2: if $\tau_{ij} = \tau_{gh}$ and $d_{ij} < c_1 \cdot d_{ij}$ then
3: return false;
4: end if
5: if $\tau_{gh} < \tau_{ij}$ and $d_{gh} < d_{gh}$ then
6: return false;
7: end if
8: if $\tau_{gh} < \tau_{ij} \leq \tau_{gh} + \frac{(1+\log b) \cdot \log n}{\alpha_{ij}}$ and $d_{hi} < c_1 \cdot d_{gh}$ then
9: return false;
10: end if
11: if $\tau_{gh} + \frac{(1+\log b) \cdot \log n}{\alpha_{ij}} < \tau_{ij}$ and $d_{hi} < n^{\frac{1}{2}} \cdot d_{ij}$ then
12: return false;
13: end if
14: end for

Property 3.6: Consider two links $l_{ij}$ and $l_{gh}$ with $\gamma_{ij} = \gamma_{gh}, it holds that $d_{ij} \geq 2^{(\tau_{gh} - \tau_{ij}) \cdot a_1} \cdot d_{gh}$ if $\tau_{ij} < \tau_{gh}$.

Property 3.7: Consider two links $l_{ij}$ and $l_{gh}$ with $\gamma_{ij} = \gamma_{gh}$, their length is either very similar or vastly different.

This property can be concluded from two sides. When the two links have the same $\tau$ value, $\frac{1}{2} \leq \frac{d_{ij}}{d_{gh}} \leq 2$ so their
lengths differ only by a factor of 2. Otherwise, when they have different \( \tau \)'s, by Property 3.6 they have lengths that differ by at least a factor of 2, possibly by an exponential amount.

**Property 3.8:** Consider two links \( l_{ij} \) and \( l_{gh} \), if \( d_{ij} > d_{gh} \), then \( \tau_{ij} \leq \tau_{gh} \).

This property gives the monotonicity of \( \tau \) which is non-decreasing when length decreases. It can be generated from the condition \( \tau \) is renewed.

**Property 3.9:** All the links in the same time slot \( L_t \) have the same \( \gamma \) value.

All links in \( L_t \) are from a set \( T_m \) where each link has \( \gamma = m \).

**Property 3.10:** The power assigned for each link \( l_{ij} \) is a non-linear function of \( d_{ij} \).

In (Thomas and Roger, 2006), it was proved uniform or linear power assignments can result in \( \Theta(n) \) complexity in the worst case. In light of this, non-linear power assignment is used to achieve better results in (Dariusz and Mariusz, 2010; Thomas and Roger, 2006; Thomas, 2007).

### 4 Correctness

In this section we give the proof that links scheduled in the same time slot from the algorithm above can transmit concurrently under both SINR and precedence constraints.

**Theorem 1:** Consider any set \( L_t \) generated by the algorithm, and for all links \( l_{ij} \in L_t \), the precedence constraint is satisfied and it holds that:

\[
P_{ij} \geq \frac{P_{ij}}{\sum_{l_{gh} \in L_t, l_{gh} \neq l_{ij}} \frac{P_{gh}}{d_{gh}^\beta}} \geq \beta \tag{1}
\]

Algorithm 1 schedules all nodes in no more than \( \lceil \log n \rceil \) rounds by Property 3.5. We should show that, in each round, the generated set \( L_t \) can satisfy both precedence and SINR constraints. The precedence satisfiability has already been shown by Property 3.4, we now prove the SINR constraint also holds based on Lemma 4.1,4.2,4.3 and 4.4 given below.

**Lemma 4.1:** Consider link \( l_{ij} \in L_t \) scheduled in time slot \( t \), the interference caused at \( v_j \) by other links \( l_{gh} \in L_t \) with \( \tau_{gh} < \tau_{ij} \) is bounded: \( I_2(v_j) \leq N_2 \cdot k \cdot b^{\gamma_{ij}} \).

**Proof:** Since \( l_{gh} \in L_t \) and \( \tau_{gh} < \tau_{ij} \), when \( l_{ij} \) is considered, by the \text{canschedule} subroutine we know \( d_{gj} \geq d_{gh} \). Fix \( \tau_{gh} \) and bound the interference caused by the links with the same \( \tau \) value. Divide the plane into rings \( R_1, R_2 \cdots R_\infty \). For each link \( l_{gh} \) in ring \( R_\lambda \); \( (2\lambda - 1)d_{gh} \leq d_{gj} < (2\lambda + 1)d_{gh} \). Since \( \tau \) is fixed, \( \tau_{gh} = \tau_{gh} \in \tau_{gh} \) and \( \tau_{gh} \) can be scheduled by the \text{canschedule} subroutine, so \( d_{gj} \geq \min \{ d_{gh}, c_1 d_{gh} \} \geq \frac{d_{gh}}{2} \). The disks of radius \( \frac{d_{gh}}{2} \) centered at each link’s sender don’t overlap, thus the number of the senders is bounded by:

\[
N_\lambda \leq \frac{\pi(2\lambda + 1 + \frac{d_{gh}}{2})^2 - \pi(2\lambda - 1 - \frac{d_{gh}}{2})^2}{\pi(\frac{d_{gh}}{2})^2} = 16 \cdot \frac{4(\frac{d_{gh}}{2} + 2)}{c_1^2} \leq \frac{32(c_1 + 4)}{c_1^2} \cdot \lambda \tag{2}
\]

The interference caused at \( v_j \) by all senders in ring \( R_\lambda \) is bounded by:

\[
I_{R_\lambda}(v_j) \leq N_\lambda \cdot k \cdot b^{\gamma_{gh}} \cdot (2d_{gh})^\alpha = 32(c_1 + 4) \cdot k b^{\gamma_{gh}} \cdot \frac{\lambda}{(2\lambda - 1)^\alpha} \leq 32 k b^{\gamma_{gh}}(c_1 + 4) \frac{1}{c_1^2} (\alpha - 1) b^{\gamma_{gh}} \tag{3}
\]

Naming the set of all links with \( \tau = \tau_{gh} \) as \( S_{gh} \) and combining all the rings, we can bound the total interference by senders in \( S_{gh} \) by:

\[
I_{S_{gh}}(v_j) = \sum_{\lambda=1}^{\infty} I_{R_\lambda}(v_j) \leq 32 k b^{\gamma_{gh}}(c_1 + 4) \cdot \frac{1}{c_1^2} \sum_{\lambda=1}^{\infty} \frac{1}{\alpha - 1} b^{\gamma_{gh}} \tag{4}
\]

Since \( 1 \leq \tau_{gh} \leq \tau_{ij} \), if we sum up all the interference \( I_{S_{gh}}(v_j) \), we get

\[
I_2(v_j) = \sum_{\tau_{gh}=1}^{\tau_{ij}} I_{S_{gh}}(v_j) \leq C_1 \cdot k \sum_{\tau_{gh}=1}^{\tau_{ij}} b^{\gamma_{gh}} = C_1 \cdot k b^{\gamma_{ij}-1} \leq 2 C_1 \cdot k b^{\gamma_{ij}-1}
\]

where \( C_1 = \frac{32 k b^{\gamma_{gh}}(c_1 + 4)}{c_1^2} \cdot \frac{1}{\alpha - 2} \) and \( N_1 = 2 C_1 \), the lemma follows. \( \square \)

**Lemma 4.2:** Consider link \( l_{ij} \in L_t \) scheduled in time slot \( t \), the interference caused at \( v_j \) by other links \( l_{gh} \in L_t \) with \( \tau_{gh} = \tau_{ij} \) is bounded: \( I_2(v_j) \leq N_2 \cdot k \cdot b^{\gamma_{ij}} \).

**Proof:** There are two types of links scheduled: links with length no less than \( d_{ij} \) and links with length less than \( d_{ij} \). For the first case, it’s clear that each link \( l_{gh} \in L_t \) has \( d_{gj} \geq c_1 d_{ij} \) by \text{canschedule}(l_{ij},L_t). Regarding the second case, any link \( l_{gh} \in L_t \) with \( d_{gh} < d_{ij} \) is scheduled after \( l_{ij} \), and \( d_{gj} \geq c_1 d_{gh} \geq c_1 d_{ij} \) holds according to Property 3.7.

Combining the two cases above shows \( \forall l_{gh} \in L_t \) with \( \tau_{gh} = \tau_{ij} \), \( d_{ij} \geq \frac{c_1 d_{ij}}{2} \). We divide the plane into rings \( R_1, R_2 \cdots R_\infty \) and for any link \( l_{gh} \) in \( R_\lambda(\lambda \geq 1) \); \( \frac{1}{2} c_1 d_{ij} \leq d_{gj} < \frac{1}{2} c_1 d_{ij} \). Now consider any two links \( l_{gh}, l_{gh'} \in R_\lambda \). It’s easy to see that \( d_{gh} \geq \frac{d_{gj}}{2} \)
min\{c_1d_{gh}, c_1d_{g'g'}\} \geq \frac{\tau_{gh}}{4} d_{ij}. Disks with radius \frac{d_{ij}}{4} centered at each sender in \( R_\lambda \) don’t overlap. Thus, we can bound the number of senders by:

\[ N_\lambda \leq \frac{\pi((\lambda + 1)^2 + \frac{1}{4})d_{ij}^2 - \pi(\frac{\lambda}{2} - \frac{1}{4})^2d_{ij}^2}{\pi(\frac{d_{ij}}{2})^2} = 16(\lambda + 1) \leq 32\lambda \]

By the triangle inequality we can deduce:

\[ d_{gj} > d_{gi} - d_{ij} \geq (\frac{c_1}{2} \lambda - 1)d_{ij} \geq \frac{c_1 - 2}{2} \lambda d_{ij} \]

Thus, the interference caused at \( v_j \) by senders in ring \( R_\lambda \) can be bounded by:

\[ I_{R_\lambda}(v_j) \leq 32\lambda \frac{k \cdot b^{\tau_{ij}}(2d_{ij})^\alpha}{(\frac{(c_1)^2}{2})^\alpha d_{ij}^\alpha} = 32 \cdot 4^\alpha \frac{b^{\tau_{ij}}}{c_1d_{ij}^\alpha} \cdot \frac{1}{\lambda^\alpha} \]

Since \( \lambda \geq 1 \), the sum of all the layers’ interference can be bounded by: \( I_2(v_j) = \sum_{\lambda=1}^\infty I_{R_\lambda}(v_j) \leq \frac{32\lambda \cdot 4^\alpha \cdot k b^{\tau_{ij}}}{(c_1)^2 \cdot 2^\alpha} \cdot \frac{1}{\lambda^\alpha} \leq \frac{32\lambda \cdot 4^\alpha \cdot k b^{\tau_{ij}}}{(c_1)^2 \cdot 2^\alpha} \cdot \frac{1}{\lambda^\alpha} \)

and the lemma follows. \( \Box \)

**Lemma 4.3:** Consider link \( l_{ij} \in L_t \) scheduled in time slot \( t \), the interference caused at \( v_j \) by other links \( l_{gh} \in L_t \) with \( \tau_{ij} < \tau_{gh} \leq \tau_{ij} + \frac{1(1+\log n)\log n}{\alpha a_1} \) is bounded by:

\[ I_3(v_j) \leq N_3 \cdot k \cdot b^{\tau_{ij} - 1} \]

**Proof:** We use the same technique employed in Lemma 4.1’s proof. First fix \( \tau_{gh} \), when \( l_{gh} \) is considered for scheduling by \textbf{canSchedule}(l_{gh}, L_t), it should fulfill the condition \( d_{gh} \geq c_1d_{ij} \) since \( \tau_{gh} > \tau_{ij} \). We divide the plane into an infinite number of rings \( R_1, R_2, \ldots, R_\infty \). For any link \( l_{g'g'} \) with \( \tau_{g'g'} = \tau_{gh} \) in ring \( R_\lambda(\lambda \geq 1) \), we have that \( c_1d_{ij} \leq d_{g'g'} \leq c_1(\lambda + 1)d_{ij} \). We can conclude that \( d_{g'g'} \geq \frac{d_{ij}}{4} \) if links \( l_{gh} \) and \( l_{g'g'} \) in \( R_\lambda \) have the same \( \tau \) value from the analysis above. Disks of radius \( \frac{d_{ij}}{4} \) centered at each sender don’t overlap. Thus the number of such senders(links) can be bounded by:

\[ N_\lambda \leq \frac{\pi((c_1 + \frac{5}{4})\lambda)^2d_{ij}^2 - \pi((c_1 - \frac{1}{4})\lambda)^2d_{ij}^2}{\pi(\frac{d_{ij}}{4})^2} \]

\[ = 24(2\lambda + 1) \frac{d_{ij}^2}{d_{gh}^2} \leq 72\lambda \frac{d_{ij}^2}{d_{gh}^2} \]

(5)

The interference caused by ring \( R_\lambda \) can be bounded by:

\[ I_{R_\lambda}(v_j) \leq 72\lambda \frac{d_{ij}^2}{d_{gh}^2} \cdot \frac{k \cdot b^{\tau_{ij}} \cdot (2d_{gh})^\alpha}{(c_1\lambda d_{ij})^\alpha} \]

\[ \leq 72k \frac{2^\alpha \cdot b^{\tau_{ij}}}{c_1^\alpha} \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \]

(6)

Combining the interference over all rings, for any set \( S_{gh} \), the interference at some fixed \( \tau_{gh} \) value can be deduced:

\[ I_{S_{gh}}(v_j) = \sum_{\lambda=1}^\infty I_{R_\lambda}(v_j) \]

\[ \leq 72k \frac{2^\alpha \cdot b^{\tau_{ij}}}{c_1^\alpha} \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \]

(7)

Taking Property 3.6 into consideration of we know that \( d_{ij} \geq 2(\tau_{ij} - \tau_{gh} - 1) \cdot d_{gh} \). Inequation 7 can be transformed into:

\[ I_{S_{gh}}(v_j) \leq 72k \frac{2^\alpha \cdot b^{\tau_{ij}}}{c_1^\alpha} \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \]

\[ \geq 72k \frac{2^\alpha \cdot b^{\tau_{ij}}}{c_1^\alpha} \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \]

since we can choose some appropriate value for \( b \) and \( a_1 \) to suit:

\[ (b^{\tau_{gh} - \tau_{ij}} \leq (2^\alpha) \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \]

(8)

The sum of all the interferences over different \( \tau \) values is then:

\[ I_3(v_j) \leq \sum_{\tau_{gh} = \tau_{ij} + 1}^\infty \frac{72k \cdot 2^\alpha \cdot b^{\tau_{ij}}}{c_1^\alpha} \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \]

\[ \leq 2 \cdot \frac{72k \cdot 2^\alpha \cdot b^{\tau_{ij}}}{c_1^\alpha} \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \]

(9)

Let \( N_3 = 2 \cdot \frac{72k \cdot 2^\alpha \cdot b^{\tau_{ij}}}{c_1^\alpha} \frac{d_{gh}^\alpha - 1}{\lambda^\alpha} \) and the interference is bounded by \( I_3(v_j) \leq N_3 \cdot k \cdot b^{\tau_{ij} - 1} \).

**Lemma 4.4:** Consider link \( l_{ij} \in L_t \) scheduled in time slot \( t \), the interference caused at \( v_j \) by other links \( l_{gh} \in L_t \) with \( \tau_{gh} \geq \tau_{ij} + \frac{1(1+\log n)\log n}{\alpha a_1} \) is bounded by:

\[ I_4(v_j) \leq N_4 \cdot k \cdot b^{\tau_{ij} - 1} \]

**Proof:** Since in \textbf{canSchedule}(l_{gh}, L_t) \( l_{gh} \) will pass line 11 and generate \( d_{gh} \geq n \cdot \frac{d_{ij}}{d_{gh}} \cdot b^{\tau_{ij} - 1} \), then the interference caused by such a single link is:

\[ I_{S_{gh}}(v_j) = \frac{k \cdot b^{\tau_{ij}} \cdot d_{gh}^\alpha}{d_{ij}^\alpha} \leq \frac{n \cdot d_{gh}^\alpha \cdot b^{\tau_{ij} - 1}}{d_{ij}^\alpha} \]

There are at most \( n \) such links, the interference caused at \( v_j \) is at most:

\[ I_4(v_j) \leq N_4 \cdot k \cdot b^{\tau_{ij} - 1} \]

\( \square \)
Lemma 4.5: For all links $l_{ij} \in L_i$ scheduled in time slot $t_i$, it holds that:

$$\frac{P_{ij}}{d_{ij}^2} \geq \beta$$

Proof: Combining Lemma 4.1, 4.2, 4.3 and 4.4:

$$\frac{P_{ij}}{d_{ij}^2} \geq \frac{N + \sum_{l_{ig} \in L_i, l_{ig} \neq l_{ij}} \frac{P_{ig}}{d_{ig}^2}}{N + \sum_{l_{ig} \in L_i} \frac{P_{ig}}{d_{ig}^2}} \geq \frac{N + (I_1(v_j) + I_2(v_j) + I_3(v_j) + I_4(v_j))}{N + (N_1 + N_2 + N_3 + 1)k \cdot b^r_{ij} - 1} \geq \beta$$

Remark 4.1: Because $N_1, N_2, N_3$ are related to $c_1$ which can be very large and $b$ which is related to $N$ and $\beta$, $k$ can be assigned to be small enough in relation to $N$ such that $SINR$ is satisfied in 11. For details, please see Section 6.

From Lemma 4.5 and Property 3.5, all sets generated in each round can hold both precedence and SINR constraints, so Theorem 1 holds.

5 Complexity Analysis

In this section, we show that Algorithm 1 can schedule all links in $O(\log^2 n)$ time slots even in worst case arbitrary deployment.

Theorem 2: Algorithm 1 can schedule all links $l_{ij} \in L_i$ in $O(\log^2 n)$ time slots.

From Property 3.5, it's obvious that Algorithm 1 will terminate in $\lceil \log n \rceil$ rounds. If we can prove all links in $T'$ can be scheduled in $O(\log n)$ time slots for each single round, then Theorem 2 is proved.

Lemma 5.1: Given a set of disks $\Gamma$ of radius $R$ no less than $R$, and a disk $U$ of radius $R$. There are at most 18 disks $d_i \in \Gamma$ such that: (1) $d_i$ intersects $U$ and (2) $d_i$ doesn't intersect the center of $d_j, j \neq i$.

More concretely, for disk $U$ with center $c_U$, try to use disks $R_i$ with radius $r_i \geq R$, center $c_i$ to intersect $U$, i.e. $d_{c_i, c_U} \leq R + r_i$. No two such disks cover any other's center, that is $d_{c_i, c_j} \geq max\{r_i, r_j\}$. Then the number of such circles can be bounded by some constant. (Bateman and Erdős, 1951) shows that setting the constant to 18 is sufficient.

Lemma 5.2: Consider all links $l_{ij} \in T'$ of length $d_{ij} \geq R$. For any disk of radius $R$, there can be at most $C$ receivers of such links in it, and $C$ is some constant number (actually it is 18).

Proof: We use Lemma 5.1 to get the result. For any disk with radius R, suppose the center is $c$ and there are links $\Gamma$ which meet the condition specified in Lemma 5.1. Then:

- For each link $l_{ij}$, the receiver must be in the circle of radius R, so $d_{c, ij} \leq R + d_{ij}$;

- For any two links $l_{ij}, l_{ig} \in \Gamma$, $d_{ig} \geq max\{d_{ij}, d_{gh}\}$ must be satisfied. By way of contradiction, suppose the inequality is not true. Then without loss of generality, suppose $d_{ig} < d_{ij}$. From Property 3.2 $l_{ig}$ must be in $T(X)$. Since $l_{ig} \in \Gamma \subseteq T'$ we have a contradiction, therefore $d_{ig} \geq max\{d_{ij}, d_{gh}\}$ holds.

From the two points above, the problem can be transferred to one related with Lemma 5.1. That is to say, the number of such links is bounded by 18, which means $|\Gamma| \leq 18$. So Lemma 5.2 follows.

Lemma 5.3: The number of disks with radius $\frac{R}{2}$ needed to cover a circle $C$ with radius $R$ completely is bounded by 9.

Proof: Consider the square with length $l = 2R$ that contains disk $C$. Divide it into 9 smaller squares of length $l' = \frac{2R}{3}$. Since the diagonal length of the smaller squares are $\frac{2R}{3} \sqrt{2} < R$, a single disk with radius $\frac{R}{2}$ is enough to cover it. So 9 disks are enough to cover the square that contains disk $C$.

In order to bound the number of time slots required to schedule all links in the Scheduling Step, we use the notion of blocking links as described below:

Definition 5.1: A link $l_{gh}$ is a blocking link for $l_{ij}$ if: $\gamma_{gh} = \gamma_{ij}$, $d_{gh} \geq d_{ij}$, and canSchedule($l_{ij}, L_i$) returns false if $l_{gh} \in L_i$. Let $B_{ij}$ denote the set of blocking links of $l_{ij}$.

Now the main task is counting the number of blocking links of each link and give a bound. If we can prove the number is bounded by $C_1 \log n$ where $C_1$ is some constant, then we can just schedule the link in $C_2 \log n + 1 \leq C_2 \log n$ time slots, where $C_2$ is also some constant. So we can get the desired $O(\log n)$ result. Let:

- $B_{ij}^\gamma$ be the set of blocking links $l_{gh} \in B_{ij}$ where $\tau_{ij} = \tau_{gh}$;

- $B_{ij}^\gamma$ be the set of blocking links $l_{gh} \in B_{ij}$ where $\tau_{ij} > \tau_{gh}$;

Since the algorithm schedule all links in decreasing order of the length in the main loop, we need not consider the case when $\tau_{ij} < \tau_{gh}$. Now we give two bounds respectively.
Lemma 5.4: For all links \( l_{ij} \in T' \), the number of blocking links in \( B_{ij}' \) is at most \( O(\log n) \).

Proof: Since each link \( l_{gh} \in B_{ij}' \) has \( \tau_{ij} = \tau_{gh} \), we know \( d_{ij} \leq d_{gh} \leq 2d_{ij} \), and from line 2 of the \texttt{canSchedule} subroutine: \( d_{ij} \leq c_{1}d_{ij} \), which means all the senders of blocking links must be in the disk of radius \( c_{1}d_{ij} \) at sender \( c_{1} \). For any two links \( l_{gh}, l_{g'h'} \in B_{ij}' \), we know that \( d_{gh'} \geq \max\{d_{gh}, d_{gh'}\} \geq d_{ij} \) by the analysis of Lemma 5.2. If we draw a disk of radius \( \frac{d_{ij}}{2} \) centered at all senders in the disk of radius \( c_{1}d_{ij} \), no two disks will overlap, and the number of the blocking links’ senders can be bounded by:

\[
N \leq \frac{\pi(c_{1} + \frac{1}{2})^{2}d_{ij}^{2}}{\pi \left( \frac{d_{ij}}{2} \right)^{2}} = 4(c_{1} + \frac{1}{2})^{2}
\]

Thus there can be at most a constant number senders satisfying the constraint, so \( |B_{ij}'| \leq 4(c_{1} + \frac{1}{2})^{2} \in O(\log n) \).

Lemma 5.5: For all links \( l_{ij} \in T' \), the number of blocking links in \( B_{ij}' \) is at most \( O(\log n) \).

Proof: We need to consider three cases in the \texttt{canSchedule} subroutine line 5, line 8 and line 11:

1. When \( \tau_{gh} < \tau_{ij} \) and \( d_{ij} < d_{gh} \) : There can be at most one satisfying link. Since \( d_{ij} < d_{gh} \) and \( l_{gh} \in T' \), \( l_{g} \) must belong to the tree by the same analysis and arrive at a contradiction.

2. \( \tau_{gh} < \tau_{ij} \leq \tau_{gh} + \frac{(1+\log b)\log n}{a_{ij}a_{gh}} \) and \( d_{hi} < c_{1}d_{gh} \) : Consider a fixed \( \tau_{gh} \) and compute the number of links with the same \( \tau_{gh} \) value. The condition \( d_{hi} < c_{1}d_{gh} \) means the receivers of such links must be located in the disk of radius \( c_{1}d_{gh} \) at \( v_i \). Now use disks with radius \( \frac{c_{1}d_{gh}}{2} \) to cover this region. From Lemma 5.3 we can use 9 such disks to cover it. Obviously each disk with radius \( \frac{c_{1}d_{gh}}{2} \) can be covered by \( c_{1}^{2} \) disks with radius \( \frac{d_{gh}}{2} \). Thus, 9\( c_{1}^{2} \) disks with radius \( \frac{d_{gh}}{2} \) can cover the region with radius \( c_{1}d_{gh} \). For any other link \( l_{g'h'} \), it can be shown that \( d_{gh'} \geq \frac{d_{gh}}{2} \).

According to Lemma 5.2, we can take at most \( C \) receivers in each little disk and there are at most \( 9Cc_{1}\) receivers satisfying the condition. Since there are \( \frac{(1+\log b)\log n}{a_{ij}a_{gh}} \) different \( \tau_{gh} \) values, at most \( \frac{(1+\log b)\log n}{a_{ij}a_{gh}} \cdot 9Cc_{1} \in O(\log n) \) receivers of such blocking links exist.

3. \( \tau_{gh} + \frac{(1+\log b)\log n}{a_{ij}a_{gh}} < \tau_{ij} \), \( d_{hi} < n^\tau d_{ij}b^{(\tau_{ij}-\tau_{gh})+1} \) : Since there are \( O(n) \) different \( \tau \) values, we can not just apply the method used above. From Property 3.6:

\[
d_{gh} \geq 2^{(\tau_{ij}-\tau_{gh})a_{1}} \cdot d_{ij} \geq 2^{a_{1}} \cdot \frac{(1+\log b)\log n}{a_{1}} \cdot d_{ij}
\]

Consider the first layer: for each link \( l_{gh} \) that satisfies the condition and \( d_{hi} < n^\tau d_{ij}b^{(\tau_{ij}-\tau_{gh})+1} \), the number of receivers of such blocking links can be bounded from Lemma 5.2 to at most \( C \) receivers in that layer. Then consider links \( l_{gh} \) with receivers such that:

\[
d_{hi} < n^\tau d_{ij}b^{(\tau_{ij}-\tau_{gh})+1}, d_{ij} > n^\tau d_{ij}b^{(\tau_{ij}-\tau_{gh})+1}
\]

We call this layer \( B_{ij}' \) for \( \varphi = 0 \). Suppose there is such a link in this layer, it must be the case that \( d_{hi} < n^\tau d_{ij}b^{(\tau_{ij}-\tau_{gh})+1} \), so:

\[
\alpha \varphi < \frac{n^\tau d_{ij}b^{(\tau_{ij}-\tau_{gh})+1}}{\log n}
\]

So we can get \( \varphi < C_{2}\log n \) for some constant \( C_{2} \) and \( \tau_{ij} < \tau_{gh}n^{\varphi} \). Thus:

\[
d_{gh} \geq 2^{(\tau_{ij}-\tau_{gh})a_{1}} \cdot d_{ij} \geq 2^{a_{1}} \cdot \frac{(1+\log b)\log n}{a_{1}} \cdot d_{ij}
\]

If we choose an appropriate value for \( a_{1} \) to meet following inequality

\[
(\alpha \varphi \cdot a_{1} - \alpha \varphi \log b - \frac{1}{\alpha}) \cdot \log n > a_{1}
\]

inequality 16 can be written as:

\[
d_{gh} \geq 2^{a_{1}} \cdot \frac{(1+\log b)\log n}{a_{1}} \cdot d_{ij}
\]

Using Lemma 5.2, for the disk of radius \( R = b^{\alpha \varphi} \log n \cdot d_{ij} \) at \( v_{i} \), there can be only a constant number of receivers with length larger than \( R \) in the disk, so there can be at most constant number of blocking links in layer \( B_{ij}' \). Since \( \varphi < C_{2}\log n \) and each layer can have at most a constant number of blocking links, the number of blocking links is bounded by \( O(\log n) \).

From the three points above, we can conclude that \( B_{ij}' \) has at most \( O(\log n) \) links.

Lemma 5.6: In each round, \( \forall l_{ij} \in T' \), \( l_{ij} \) can be scheduled in time slot \( 0 \leq t(i_{ij}) \leq C\log(n) \) for some constant \( C \).

Proof: From Lemma 5.4 and Lemma 5.5 we deduce that the number of \( l_{ij} \)’s blocking links is bounded by \( O(\log n) \), so each link can be scheduled in \( O(\log n) \) time slots. Since \( \gamma \) ranges from 1 to \( a_{1} \) in Algorithm 4, Line 2 when \( a_{1} \) is a constant to be chosen, all links can be scheduled in \( O(\log n) \) time slots, so Lemma 5.6 follows.

Remark 5.1: After constructing the topology using nearest neighbor tree method, we can generate a \( O(\log n) \) schedule for the Connectivity Problem by directly adopting the subroutine Algorithm 4.
6 Parameter Constraints

There are 4 parameters $a_1, b, k$ and $c_1$ in our algorithm. The constraints of the parameters should be satisfied:

- In the proof of Lemma 4.3, Inequation 8 should be satisfied, so: $b^2 \leq 2^{a_1} \Rightarrow a_1 \geq 2\log b$;
- In the proof of Correctness, Inequation 11 should be satisfied, thus:
  $$k \cdot b^{\tau_{ij}} \geq \beta \cdot [N + (N_1 + N_2 + N_3 + 1)k \cdot b^{\tau_{ij} - 1}]$$
  Letting $c_1$ be a very large value lets
  $$N_1 + N_2 + N_3 < 1,$$
  and so
  $$k \cdot b^{\tau_{ij}} \geq \beta \cdot (N + 2kb^{\tau_{ij} - 1}).$$
  This can be rewritten as:
  $$k(b - 2\beta) \cdot b^{\tau_{ij} - 1} \geq \beta N.$$ Since $\tau_{ij} \geq 1,$
  $$k(b - 2\beta) \geq \beta N$$ should be satisfied.
- In the proof of Lemma 5.5, Inequation 17 should be fulfilled:
  $$(a^\varphi \cdot a_1 - a^\varphi \log b - \frac{1}{\alpha}) \cdot \log n > a_1;$$

Here is an example, $b = 2\beta + \beta N,$ $k = 1,$ $a_1 = \lfloor 2\log b + \frac{1}{\alpha} \rfloor + 1,$ and $c_1$ be a very large value. This shows that such parameters can be assigned easily to support the algorithm.

Remark 6.1: $c_1$ is a not just very large value, it is actually related to $b$ and $\alpha$. Here we give a bound to satisfy Inequation 11: $c_1 > 2 + \frac{a_1 - 1}{\alpha - 2} \cdot (272 + 3262\alpha)2^\alpha$.

7 Conclusion and Future Work

This paper proposes an algorithm which solves the MLAS problem under the SINR model. By rationally combining several techniques for wireless network scheduling, like round scheduling, topology construction and non-linear power assignment, our algorithm always produces a feasible aggregation scheduling policy with latency bounded by $O(\log^2 n)$ time slots. To the best of our knowledge, this is the best solution of this problem to date. Compared with previous works (Chen, Hu, and Zhu, 2005; Huang et al., 2007; Wan et al., 2009; Xu et al., 2009; Yu, Li, and Li, 2009) under protocol models, our algorithm gives more instructions for real world applications in wireless sensor networks, because we adopt the physical model which is a much better description of reality, even though our algorithm still suffers from being a centralized one. Our algorithm also gives a better result than all the previous works (Li et al., 2009; Li et al., 2010) on the same problem under the SINR model. Moreover, the subroutine of our algorithm can generate a $O(\log n)$ schedule for Connectivity Problem.

In the future, we will focus on improving the efficiency of this centralized algorithm and designing an efficient distributed one. The theoretical result analyzed in this paper shows that the exact implementation of such an algorithm would be cumbersome, as the non-linear power assignment are non-trivial. More simple algorithms which are easy to use in real applications may be a good research point as well, even if they would probably suffer from loss of efficiency.

8 Acknowledgement

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Foundation of China Grant 61103186, 61073174, 61033001, 61061130540 and the Hi-Tech research and Development Program of China Grant 2006AA10Z216.

References


