Optimal Packet Scheduling for Steerable Directional Antenna Systems

Chi-Yao Hong∗, Ai-Chun Pang†
∗†Department of Computer Science and Information Engineering
‡Graduate Institute of Networking and Multimedia
National Taiwan University, Taipei, Taiwan, R.O.C.
Email: ∗cyhong@newslab.csie.ntu.edu.tw, †ac pang@csie.ntu.edu.tw

Abstract

In this study, we focus on the packet scheduling problem especially with respect to the steering overhead of the steered antenna system. Our scheduling problem is investigated with a generic network model, while the objective of the problem is the throughput metric, which is the essentially metric in wireless systems. We prove the tractability of the problem by proposing an optimal covering algorithm (CA) in time \(O(|S|^2)\) where \(|S|\) is the number of the neighbors (i.e., the connectivity). Based on the result of CA, we further proposed an optimal ordering algorithm (OA) in time \(O(|S| \ln |S|)\) to minimize the maximum lateness of the traffic demand of its neighbor.

I. INTRODUCTION

Antennas play a significant role in next generation wireless networks. The smart antenna technologies are arising in recent decade as a promising solution for future wireless systems. Especially, the steering of nulls reduces the interference of the jammer, and potentially further improves the spatial spectrum reusing. Furthermore, the adoption of the directional beam antennas advantages the signal sensitivity for the desired direction, which could benefit the system capacity. While the use of directional antennas brings the better performance and alleviates the signal fading as compared to the use of omni-directional antennas, one known disadvantage is the lack of mobility support since the narrow beamwidth of directional antennas limits the antenna’s coverage area. To overcome such drawback of the high gain directional antenna, the electronically steerable
antennas are developed. Through the estimation of direction of arrival (DOA) techniques [1], the location of the target can be tracked. Then the main beam can be adaptively switched into the proper direction in order to ensure the ubiquitous coverage.

To realize the potential for high-speed next generation networks, the MAC design with directional beam antennas have been extensively investigated in ad hoc networks [2], [3], wireless sensor networks [4] and vehicular networks [5]. In [2], a novel MAC protocol exploiting the directional antennas is presented for static ad hoc networks. Sundaresan et al. [3] proposed an efficient link scheduling algorithm with digital adaptive array (DAA) antennas. Although the beam steering/switching is flexible to accommodate the mobility of wireless devices, frequently steering/switching of the main beam’s direction leads serious performance degradation since the steering/switching delay ranges 250 to 1200 microseconds in practical [5], [6]. However, very little work has been done to consider such overhead. Specifically, in [6], the authors showed that the steering/switching delay, which constrained the system throughput, becomes the performance bottleneck especially in high-speed networks. Navda et. al implemented a novel beam steering framework for vehicular network access [5].

In this study, we focus on the packet scheduling problem especially with respect to the steering overhead of the steered antenna system, which is lack of consideration in previous study to the best of our knowledge. As compared to the switched beam antennas (which has been considered in [6]), the steerable antennas provide higher scalability and fine-granularity of steering. Our scheduling problem is investigated with a generic network model, while the objective of the problem is the throughput metric, which is the essentially metric in wireless systems. We prove the tractability of the problem by proposing an optimal covering algorithm (CA) in time $O(|\mathcal{S}|^2)$ where $|\mathcal{S}|$ is the number of the neighbors (i.e., the connectivity). Based on the result of CA, we further proposed an optimal ordering algorithm (OA) in time $O(|\mathcal{S}| \log |\mathcal{S}|)$ to minimize the lateness.

The remainder of the paper is organized as follows. Section II formally presents our problem formulation. In Section III, we present our packet scheduling algorithm, and the optimality proof is also presented in this section. Finally, the paper is concluded with a brief summary and some pointers to future work.
II. Problem Definition

This paper considers a wireless station $Y$ equipped with a steerable antenna. Let $\mathcal{S} = \{s_0, s_1, \ldots, s_{|\mathcal{S}|-1}\}$ be the set of $Y$’s neighbors $s_i$. Each neighbor $s_i$ is associated with a service time $p_i$ and a delay requirement $d_i$. Consider that the station $Y$ spans a disk, then the corresponding plane angle of each neighbor $s_i$ is denoted by $\theta_i$ degree ($\pi/180$). The directional antenna is also associated with an origin angle $\phi$ degree. This represents that the antenna aims at degree $\phi$ to $\phi + \varphi$ degree, where $\varphi$ is the horizontal beamwidth degree. Suppose that the antenna with some origin angle covers a subset of $\mathcal{S}$. Then the collection of the set of covered neighbors of the antenna with different origin angle can be defined as $\mathcal{C} = \{C_0, C_1, \ldots, C_{|\mathcal{C}|-1}\}$ where $C_j = \{s_j | s_j \in \mathcal{S}; \phi \leq \theta_j < \phi + \varphi\}$. Throughout this paper we call $C_j$ a cover. Figure 1 illustrates an example of the system model.

The time satisfying the traffic demand of any station in $\mathcal{S}$ can be written as

$$\sum_{s_i \in \mathcal{S}} p_i + \zeta \times t_s$$

where $\sum_{s_i \in \mathcal{S}} p_i$ is the overall transmission latency. We use $t_s$ and $\zeta$ to denote the beam switching latency and the number of beam switching, respectively. It is clear that the system throughput can be optimized if $\zeta$ is minimized. This gives rise to a covering problem, i.e., using minimal beam switching to cover all neighbors. Specifically, let a set cover $\mathcal{C}' \subseteq \mathcal{C}$ be a family of $C_j$. Then the problem of maximizing the throughput is then formulated as:

$$\min |\mathcal{C}'|$$

subject to

$$\bigcup_{C_j \in \mathcal{C}'} C_j = \mathcal{S}$$

The problem is similar to the Minimum Set Cover, which is commonly known as an $NP$-hard problem. In Minimum Set Cover, we are given a collection $\mathcal{L}$ of subsets of a finite set $\mathcal{F}$. The goal is find a minimal set cover $\mathcal{L}' \subseteq \mathcal{L}$ such that every element in $\mathcal{F}$ belongs to at least one member of $\mathcal{L}'$. The problem complexity will be elaborated in Section III.

After $\mathcal{C}'$ is determined, we could obtain $\zeta = |\mathcal{C}'| - 1 \geq 0$ if the neighbors in the same cover (i.e., $s_i \in C_j \in \mathcal{C}'$) will be served without switching. Nevertheless, we observed that the serving order of the cover of antenna and that of the neighbors within each cover have no effect on throughput but delay. To minimize the effect of the delay of each demand, the scheduling order should be properly designed. In order to exhibit the problem, we first define some variables as
following. Let \( C' \) be re-indexed in some serving order, namely *inter-cover sequence*. For each \( C'_{j} \in C' \), letting \( S_{j} \) be the ordered set of neighbors \( s_{i} \in C'_{j} \) in some serving order, namely *intra-cover sequence*. The *starting time* of \( s_{i} \), denoted by \( \xi_{i} \), is the schedule instant at which neighbor \( s_{i} \in C'_{j} \) starts to serve its demand. Based on the order of \( C' \) and \( S_{j} \), the starting time \( \xi_{i} \) can be derived by

\[
\xi_{i} = \left( \sum_{j' < j; C'_{j'} \in C'} \sum_{s \in S_{j'}} p_{i} \right) + (j - 1) \times t_{s} + \sum_{j' < i; s \in S_{j'}} p_{i}
\]

and the lateness of demand \( i \), denoted by \( \psi_{i} \), is

\[
\psi_{i} = \xi_{i} + p_{i} - d_{i}
\]

Then our second problem is seeking for the serving order, including *inter-cover sequence* and *intra-cover sequence*, such that the maximum lateness \( \max_{s \in S} \{\psi_{i}\} \) is minimized.

III. Our Proposed Algorithms

This section presents our proposed algorithms. First, for the covering problem we propose a fully polynomial time covering algorithm, namely \( CA \), and present the optimality analysis of \( CA \). To further improve the tardiness of the demands, an optimal ordering algorithm (\( OA \)) minimized the maximum lateness is then proposed while the optimality proof is also presented.

A. Covering Algorithm (\( CA \))

To solve the covering problem, the divide-and-conquer strategy is adopted by \( CA \). Without loss of generality, we assume that each \( C_{j} \) in \( C \) is nondecreasing with respect to \( \phi \), while the \( s_{i} \in S \) is nondecreasing with respect to \( \theta \). The basic idea of our \( CA \) is explained as follows. We divide the problem into \( |X| \) subproblems where \( X_{i} = \{C_{j}; s_{i} \in C_{j}; C_{j} \in C\} \) for each \( s_{i} \in S \) and \( X \) is the smallest-size \( X_{i} \). The subproblem assumes that \( C_{j} \) should belong to \( C' \) for each \( C_{j} \) in \( X \). Suppose that \( C_{j} \) is included in \( C' \) of the optimal solution. Then the covering problem becomes finding a subset of \( C \setminus \{C_{j}\} \) such that every neighbors in \( S \setminus \{C_{j}\} \) belongs to at least one cover in the subset. Note that the neighbors in \( S \setminus \{C_{j}\} \) are within the counterclockwise angle from \( \phi_{j} \) to \( (\phi_{j} + \varphi_{j}) \). For each subproblem, \( CA \) followed most-uncovered-first greedy method which chooses repeatedly, as its name implies, the \( C'_{j} \) which included most unselected neighbors through the counterclockwise angle.
Algorithm 1 shows the elaboration of the proposed CA. In each iteration (say, $j$th iteration) of the for loop of lines 1-6, we call Algorithm 2 to compute the minimal-size covering $C^\dagger \subseteq C$ under the hypothesis that $C_j$ should be selected. Lines 3-4 updates the best $C^\dagger$ into $C'$, which has smallest size up to the present. In Algorithm 2, a cover set $C^\ddagger$ is initially set to $\{C_j\}$ as a working temporal. Since $s_i \in S$ are indexed in nondecreasing order of its $\theta$, we have $C_j = \{s_i, s_{i+1}\%|S|, \ldots, s_{i+|C_j|-1}\%|S|\}$ for some $i$. Let $c_{\_index}$ be the index of $C$. It represents $C_{c_{\_index}}$ are currently considered to be selected into $C^\ddagger$. Since $C_j$ have to be selected, the $c_{\_index}$ is initially set by $(j+1)\%|C|$. To assure that each cover is taken into account through the counterclockwise angle, $c_{\_index}$ ranges $(j+1)\%|C|, (j+2)\%|C|, \ldots, (j+|C|-1)\%|C|$. Letting $s_{\_index}$ be the index of $S$ that signifies $S_{s_{\_index}}$ are considered to be covered. In order to assure that each neighbor is taken into account within the counterclockwise angle, $s_{\_index}$ ranges $(i + |C_j|)\%|S|, (i + |C_j| + 1)\%|S|, \ldots, [i + |C_j| + (|S| - |C_j| - 1)]\%|S|$. The for loop of Lines 4-10 and 11-17 respectively update $\phi_i$ and $\theta_i$ into $\hat{\phi}_i$ and $\hat{\theta}_i$ such that $\hat{\phi}$ and $\hat{\theta}$ accords with the nondecreasing order through the counterclockwise angle. $q_i$, initially set by 0, is a flag indicating whether $s_i$ is to be covered. The for loop of lines 18-29 repeatedly finds a cover $C_{c_{\_index}}$ covers most uncovered neighbors including $s_{s_{\_index}}$, i.e., the cover $C_{c_{\_index}}$ with maximal $\phi_{c_{\_index}} < \theta_{s_{\_index}}$.

Algorithm 1 CA

Require: $C$, $S$

Ensure: $C'$

1: for all $C_j \in X$ do
2: \hspace{1em} $C^\dagger \leftarrow \text{CA\_Greedy}(C, C_j, S)$
3: \hspace{1em} if $|C^\dagger| < |C'|$ then
4: \hspace{2em} $C' \leftarrow C^\dagger$
5: \hspace{1em} end if
6: end for
7: return $C'$

Since the running time of the for loop of lines 4-9, 11-17, and 18-28 are respectively $O(|C|)$, $O(|S|)$, and $O(\max(|C|, |S|))$, the time complexity of Algorithm 2 is $O(\max(|C|, |S|))$, dominated by updating $c_{\_index}$ and $s_{\_index}$ by an incremental manner. In Algorithm 1, there are $|X|$ calls
Algorithm 2 CA_Greedy

Require: $C, C_j, S$

Ensure: $C^\dagger$

1: Let $C_j = \{s_i, s_{(i+1)\%|S|}, \ldots, s_{(i+|C_j|-1)\%|S|}\}$
2: $C^\dagger \leftarrow \{C_j\}$
3: $c_{\text{index}} \leftarrow (j + 1)\%|C|$
4: for all $C_{j'} \in C$ do
5: if $j' < j$ then
6: $\hat{\phi}_{j'} \leftarrow \hat{\phi}_{j'} + 360^\circ$
7: else
8: $\hat{\phi}_{j'} \leftarrow \hat{\phi}_{j'}$
9: end if
10: end for
11: for all $s_r \in S$ do
12: if $i' < i$ then
13: $\hat{\theta}_r \leftarrow \theta_r + 360^\circ$
14: else
15: $\hat{\theta}_r \leftarrow \theta_r$
16: end if
17: end for
18: for $s_{\text{index}} = (i + |C_j|)\%|S|$ to $|(i + |S| - 1)\%|S|$ do
19: if $\varrho_{s_{\text{index}}} = 0$ then
20: while $\hat{\phi}_{c_{\text{index}}(c_{\text{index}}+1)\%|C|} < \hat{\theta}_{s_{\text{index}}}$ do
21: $c_{\text{index}} \leftarrow (c_{\text{index}} + 1)\%|C|$
22: end while
23: insert $C_{c_{\text{index}}}$ into $C^\dagger$
24: for all $s_{\ell} \in C_{c_{\text{index}}}$ do
25: $\varrho_{\ell} \leftarrow 1$
26: end for
27: end if
28: end for
29: return $C^\dagger$
of Algorithm 2. The total running time of CA, therefore, is $O(|X| \max(|C|, |S|))$

**Corollary 1:** $|X| \leq |C|$

**Theorem 1:** $|C| \leq |S|$

**Proof:** Without loss of generality we assume that $C_j \notin C_{j'}$ for any $j \neq j'$. If $C_j \subseteq C_{j'}$, we could simply remove $C_j$ from $C$ without causing any performance degradation because $C_{j'}$ can fully cover $C_j$. We denote the first $s_i$ of each $C_j \in C$ as $s(C_j)$. Now we prove that theorem by contradiction. Suppose that $|C| > |S|$. By the Pigeonhole Principle, there exist $C_j$ and $C_{j'}$ such that $s(C_j) = s(C_{j'})$. It suffices to show that either $C_j \subseteq C_{j'}$ or $C_{j'} \subseteq C_j$ holds, and we reach the contradiction.

**Corollary 2:** The time complexity of CA is $O(|S|^2)$

**Theorem 2:** CA is an optimal algorithm for throughput maximization

**Proof:** We shall show that Algorithm 2 (CA_Greedy) computes $C'$ of minimum size such that $\bigcup_{C_j \in C'} C_j = S$ under the hypothesis that $C_j$ should be selected. Without loss of generality, we defined that $\tilde{C} = C \setminus \{C_j\}$ and $\tilde{S} = S \setminus C_j$ are indexed through the counterclockwise angle from $\phi$ and $\theta$, respectively. Let $\tilde{C}^*$ be the optimal subset of $\tilde{C}$ with minimum size for covering $\tilde{S}$. The optimal substructure property that CA exploits is the following: letting $y$ be the minimum index of $\tilde{C}^*$, then $\tilde{C}^{**} = \tilde{C}^* \setminus \{C_j\}$ must be an optimal solution for $\tilde{S} \setminus \{C_j\}$. To prove it, we shall first show that $\bigcup_{C_j \in \tilde{C}^{**}} C_j = \tilde{S} \setminus \{C_j\}$. If $\tilde{C}^{**} \neq \emptyset$, it trivially holds; otherwise if $\tilde{C}^{**} = \emptyset$, we have $\tilde{S} \setminus \{C_j\} = \emptyset$ because $\tilde{C}^*$ is a covering subset of $\tilde{C}$. Suppose by contradiction that $\tilde{C}^{**}$ is not the optimal solution for $\tilde{S} \setminus \{C_j\}$. By the cut-and-paste argument, $\tilde{C}^*$ can not be the optimal solution for $\tilde{S}$. This proves the theorem.

**B. Ordering Algorithm (OA)**

In this section we study the effects of the schedule sequence on the maximal tardiness. In order to demonstrate this further, we start from stronger assumption that each $C_j \in C'$ consists of only one element (neighbor of station $Y$). Under the assumption, we claim that the problem of tardiness minimization can be solved by Jackson’s rule – sequencing the jobs in nondecreasing order of $d_j$.

**Theorem 3:** Jackson’s rule minimizes the maximal tardiness under the assumption.

**Proof:** Suppose by contraction that the optimal sequence $C^* = \{C^*_0, C^*_1, \ldots, C^*_{|C|−1}\}$ does not follow the Jackson’s rule. Then there must exist $s_i \in C^*_j$ and $s_f \in C^*_{j'}$ such that $j < j'$ while
\( d_i > d_{i'} \). We prove that \( C^* \) can not be optimal by following two cases.

1) Case 1: \( j' - j > 1 \)

There must exists an \( C^*_k \in C^* \) (\( j \le k < j' \)) such that \( s_l \in C^*_k, \ s_{l'} \in C^*_k+1, \ d_l > d_{l'} \). The theorem is proved by Case 2.

2) Case 2: \( j' - j = 1 \)

We have

\[
\xi_i = \left( \sum_{p' < j, \in C^*} \sum_{s_i' \in S} p_{i'} \right) + (j - 1) \times t_s \quad (6)
\]

\[\xi_{i'} = \xi_i + p_i + t_s > \xi_i \quad (7)\]

Consequently, the lateness of demand \( i \) and \( i' \), \( \psi_i \) and \( \psi_{i'} \), are respectively

\[
\psi_i = \xi_i + p_i - d_i \quad (8)
\]

\[
\psi_{i'} = \xi_{i'} + p_{i'} - d_{i'} = \xi_i + p_i + t_s + p_{i'} - d_{i'} \quad (9)
\]

We observe that \( \psi_i < \psi_{i'} \) because \( d_i > d_{i'} \). To show that this sequence can not be optimal, we switch the order between \( s_i \) and \( s_{i'} \). Note that the lateness of the other neighbors (\( i.e., \ \forall s_k \in \mathbb{S} \setminus \{s_i, s_{i'}\} \)) will not change. However, the starting time of demand \( i' \) and \( i \), denoted by \( \check{\xi}_{i'} \) and \( \check{\xi}_i \), now becomes \( \xi_i \) and \( \xi_i + p_{i'} + t_s \). Thus, the lateness of demand \( i' \) becomes

\[
\check{\xi}_{i'} + p_{i'} - d_{i'} = \check{\xi}_i + p_{i'} - d_{i'}
\]

\[
= \psi_{i'} - (p_i + t_s + p_{i'} - d_{i'}) + p_{i'} - d_{i'}
\]

\[
= \psi_{i'} - p_i - t_s
\]

\[
< \psi_{i'} \quad (10)
\]

Moreover, the lateness of demand \( i \) becomes

\[
\check{\xi}_i + p_i - d_i = \xi_i + p_{i'} + t_s + p_i - d_i
\]

\[
< \xi_i + p_{i'} + t_s + p_i - d_{i'} = \psi_{i'} \quad (11)
\]

Now we relax the assumption. OA prioritizes the \textit{intra-cover sequence} based on Jackson’s rule, \( i.e., \) scheduling \( s_i \in C_j \) in nondecreasing order according to \( d_i \) values for each \( C_j \in C' \). The
optimality of inter-cover sequencing can be proved by Theorem 3 since it represents a special case that $t_s = 0$. For inter-cover sequencing problem, the problem is constrained to find a proper order of each cover. Since every neighbor’s demand should be served contiguously without switching in a cover, the demand of each cover can be seen as a “packing” of the demand of each neighbor in the cover. Clearly, the service time of each cover can be signified as $P_i = \left( \sum_{i \in C_j} p_i \right) + t_s$. If we can find the delay requirement of each cover to represent the urgent level for the neighbors within the cover such that “the lateness of the cover” is the maximum lateness of the neighbors within the cover, the problem can be solved by Jackson’s rule. Specifically, letting $D_j$ and $\Psi_j$ be the delay requirement and the lateness of $S_j \in C'$, our goal is seeking for a proper definition of $D_j$ such that $\Psi_j = \max_{i \in S_j} \{ \psi_i \}$. Let $\varphi_j = \min_{i \in C_j} \{ \xi_i \}$ be the starting time of each cover. For each $s_i \in S_j \in C'$, Eq. (4) can be rewritten as

$$\xi_i = \varphi_j + \sum_{p' \leq i, s_r \in S_j} p_{r'}$$

Substituting Eq. (12) into Eq. (5), we obtain

$$\psi_i = \varphi_j + \left( \sum_{p' \leq i, s_r \in S_j} p_{r'} \right) - d_i$$

Moreover, we have

$$\Psi_j = \varphi_j + \left( \sum_{i \in S_j} p_i \right) - D_i$$

Let $\Psi_j \triangleq \max_{i \in S_j} \{ \psi_i \}$, substituting Eq. (13) into Eq. (14),

$$D_i = \left( \sum_{i \in S_j} p_i \right) - \max_{i \in S_j} \left\{ \left( \sum_{p' \leq i, s_r \in S_j} p_{r'} \right) - d_i \right\}$$

$$= \min_{i \in S_j} \left\{ d_i - \sum_{p' \leq i, s_r \in S_j} p_{r'} \right\}$$

IV. Conclusions

With rapid development of the steered antenna technology, the importance of efficient packet scheduling in steered antennas system has been underlined. To address this problem, in this paper we propose a covering algorithm (CA) in quadratic time with respect to the number of the neighbors to maximize the system throughput. Based on the result of CA, we further proposed an ordering algorithm, which aim at minimizing the maximum lateness of the traffic demands.
To evaluate the performance of our proposed algorithms, the optimality proofs are shown. For the future research, we plan to generalize our scheduling algorithms to apply with the multihop wireless networks.

References


