Reducing Feedback Overhead in Opportunistic Scheduling of Wireless Networks Exploiting Overhearing

Seung Jun Baek[1]
[1] College of Information and Communications, Korea University
136-713, Anam-dong, Seoul, Korea
[e-mail: sjbaek@korea.ac.kr]
*Corresponding author: Seung Jun Baek

Received December 10, 2011; revised January 12, 2012; accepted January 27, 2012; published February 28, 2012

Abstract

We propose a scheme to reduce the overhead associated with channel state information (CSI) feedback required for opportunistic scheduling in wireless access networks. We study the case where CSI is partially overheard by mobiles and thus one can suppress transmitting CSI reports for time varying channels of inferior quality. We model the mechanism of feedback suppression as a Bayesian network, and show that the problem of minimizing the average feedback overhead is NP-hard. To deal with hardness of the problem we identify a class of feedback suppression structures which allow efficient computation of the cost. Leveraging such structures we propose an algorithm which not only captures the essence of seemingly complex overhearing relations among mobiles, but also provides a simple estimate of the cost incurred by a suppression structure. Simulation results are provided to demonstrate the improvements offered by the proposed scheme, e.g., a savings of 63-83% depending on the network size.

Keywords: Channel state information, feedback, scheduling, Bayesian network, graphs

This research was supported in part by KEIT Contract No. 10035213-2010-01.

DOI: 10.3837/tiis.2012.02.009

1. Introduction

In this paper we study an approach to reduce the overhead associated with channel state information (CSI) feedback in wireless access networks exploiting spectral and temporal opportunism. Previous approaches have focused on fully decentralized setups where CSI at each node is not available to any other user when feedback is transmitted. In this work we study the case where the system CSI is partially observable by users prior to feedback, e.g., certain subsets of users can overhear each other's CSI feedback. The following describes how users' CSI can be shared among a subset of users. For concreteness we focus on a wideband access network adopting OFDM, e.g., 802.11n. We refer to the smallest unit of resource consisting of subcarriers allocatable to an individual user as a resource block (RB). We consider an access point (AP) and a set of nodes within its coverage area. If the AP intends to transmit to a group of nodes, it requests CSI for all the RBs for each node in the group. Consider a subset of nodes which are within transmission range of each other, say nodes A and B. Suppose A has provided CSI for all of the RBs. If B overhears the CSI fed back from A it can choose to transmit CSI only for the RBs which have 'superior' SNR values to those of A since all 'inferior' RBs would in principle be ignored upon reception at the AP. We study a scheme where each node makes binary decisions on whether or not to send feedback by comparing its RBs' CSI to those previously overheard.

An example is depicted in **Fig. 1**. Assume SNR takes values from $\{1, 2, \ldots, 9\}$. There are four nodes a, b, c and d. Nodes a, b and c are adjacent to one another where d is adjacent only to c. The order of reporting has been fixed to $a \rightarrow b \rightarrow c \rightarrow d$. Firstly a transmits feedback for all the RBs. Upon overhearing a's feedback, b transmits feedback only for RBs that have a higher SNR than those of a: the SNR values of 1st and 4th RBs are suppressed. Similar rules apply for nodes b and c. As for node d since it cannot overhear previously transmitted SNR values at the 1st, 5th and 6th RBs by a and b, and because c is silent with respect to those RBs, d is forced to transmit feedback for those RBs although they turn out to be inferior to those reported earlier to the AP. Let us define the overhead as the total number of bits fed back to the AP. If we assume it costs α bits to report the SNR of an RB, the overhead of the feedback exploiting overhearing in Fig. 1 is 17α as opposed to 24α in the full feedback case.

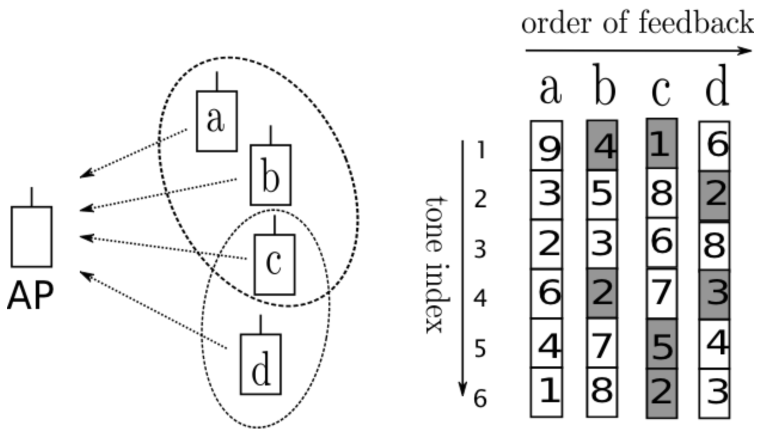

Fig. 1. An example feedback suppression procedure. The left shows the overhearing relations: nodes a, b and c are neighbors of one another where d is adjacent only to c. The right shows the feedback reports by the four nodes. Each number represents the SNR for the corresponding RB index.

The shaded numbers represent the suppressed feedback for the corresponding RBs, e.g., the report of Node b contains only four SNR values.

Our goal is to minimize the average total number of bits fed back to the AP, i.e., the average overhead. The total number of RBs per transmission time slot is denoted by *m*. Assume that the AP has infinitely backlogged queues serving a total of *n* users. Suppose under a feedback suppression scheme the probability of feedback transmission from user *k* for RB *i* is denoted by p_{ik}, $1 \leq k \leq n$ and $1 \leq i \leq m$. Assuming w_k bits are required to describe CSI of an RB for user *k*, the objective is to minimize

$$\sum_{k=1}^{n} \sum_{i=1}^{m} w_k p_{ik}. \tag{1}$$

Note w_k represents different numbers of bits per transmission required in describing SNR information/channel gains for each user capturing the heterogeneities among users, e.g., capabilities for different user equipment. For instance low-end devices may have a limited choice in the available modulation and coding schemes compared to high-end devices, thus would need less number of bits to describe CSI.

The key problem addressed in this paper is finding the optimal order in which nodes provide feedback to the AP. We show that the relationships of suppression among nodes induced by a given feedback order can be represented as a directed acyclic graph (DAG). We assign a set of random variables (RVs) associated with the event of feedback transmission to each vertex of the DAG, from which we model the system as a Bayesian network (BN) [1]. The BN encodes the joint probability distribution of the feedback transmissions for a given feedback order, and computing (1) corresponds to a marginalization of such joint distribution. Unfortunately such marginalization problems are NP-hard in general [1]. Recognizing the intractability of the problem we propose an approximation algorithm. We first show that (1) can be computed efficiently if the DAG is singly connected. A DAG is singly connected, or a polytree, if there exists at most one directed path between any two nodes. We identify a class of DAGs which have more complex structure however allow an efficient computation of the associated overhead. Based on these findings we propose an algorithm called Greedy Quantile-Based Sequential Feedback (G-QBSF) which captures the essence for significant savings in overhead. We show the efficacy of the proposed algorithm via simulation.

There has been a substantial amount of research devoted to feedback overhead reduction in wireless access networks using adaptive OFDMA, e.g., [2][3][4]. Notably using opportunistic splitting one can achieve a constant overhead in terms of the average time consumed for feedback [5] however the scheme involves coordination overhead. The work in [6], and then [7] propose schemes to eliminate coordination overheads incurred at the basestation through optimizing thresholds associated with contention probability. None of the above mentioned work, however, addresses the possibility of leveraging overhearing users' CSI to reduce overheads.

Given the SNR value of a neighbor's RB, how does a node decide if its RB quality is 'better' than its neighbors'? If the decision were to be based on the absolute SNR values, the heterogeneity in channel conditions may cause rate starvation to some nodes. As a fair resource allocation to users with heterogeneous fading statistics we adopt max-quantile (MQ) scheduling which was studied in [8][9][10]. The *quantile* is defined to be the instantaneous SNR evaluated at the distribution function of the SNR. The idea behind MQ scheduling is to select a user with the highest rate relative to its *own* distribution. Thus MQ scheduling is fair in terms of opportunism even when the channel distributions of the users are heterogeneous. Also the MQ scheduling is known to maximize the sum throughput for asymptotically large number of users, and to be robust to measurement errors in CSI [10]. In our paper we apply the MQ scheduling to both time and frequency resources, i.e., for each RB, the AP schedules a user with the maximum quantile.

This paper is organized as follows: In Section 2 we describe the system model and the proposed scheme. The problem formulation using a BN model is discussed in Section 3. Section 4 characterizes computationally tractable structures, then a greedy heuristic leveraging such structures is proposed. In Section 5 we discuss measurement distortions which can occur in our model. Section 6 presents simulation results, and Section 7 concludes the paper.

2. The Proposed Scheme

We assume a block-fading model where the channel gain of an RB is fixed during a common coherence time T_c. The channel gain is renewed in an i.i.d. manner every T_c time units and the AP requests CSI feedback at the beginning of each coherence time. We assume that the SNR distributions are independent, not necessarily identical, across users and RBs. The SNR for user *i* and RB *j* has the same distribution as a continuous RV Γ_i^j with a continuous distribution function $F_{\Gamma_i^j}(\cdot)$. The quantile of an instantaneous SNR γ of jth RB for user *i* is defined to be $F_{\Gamma_i^j}(\gamma)$. MQ scheduling for a given RB operates as follows: in the time duration $(k-1)T_c < t \leq kT_c$ for any integer *k*, and some RB index *j* where $1 \leq j \leq m$, the following user is scheduled:

$$i_j^*(t) = \underset{i=1,\ldots,n}{\operatorname{argmax}} \left[F_{\Gamma_i^j} \left(\gamma_i^j(t) \right) \right]$$

where $\gamma_i^j(t)$ is the realization of the SNR of RB *j* for user *i* at time *t*, and $F_{\Gamma_i^j}(\cdot)$ denotes the distribution function of Γ_i^j. Since $F_{\Gamma_i^j}(\cdot)$ is continuous, $F_{\Gamma_i^j}(\Gamma_i^j)$ is uniformly distributed on $[0,1]$.

In order to compute the quantile of an overheard SNR value, each node needs to know the distributions of SNR of its neighboring nodes. Hence we assume each node keeps track of the SNR distribution of the RBs of its neighbors. A node 'measures' the SNR of a particular neighbor by sampling the overheard SNR values transmitted by the neighbor. A node estimates the SNR distribution based on the *empirical* distribution of the SNR measurements. For such method, it has been shown in [10] that the storage overhead associated with measurements of channel distributions grows only *linearly* with the number of neighbors (and RBs). However there are two issues: firstly, since the nodes selectively transmit feedback, i.e., poor SNR values are likely to be suppressed often, the overheard information tends to be biased. We study how to correct such bias in Section 5. Secondly, for dense networks, it can be burdensome to measure SNR values and track SNR distributions of all the neighbors. Hence we propose an alternative scheme with reduced measurement overheads in Section 6.

We model the overhearing relation among the nodes by an undirected graph $\mathcal{G} = (V, E)$ where $V = \{1, 2, \ldots, n\}$ and $E \subseteq V \times V$ denote the set of vertices and edges respectively. An edge $e \in E$ between two nodes captures the fact that these nodes can overhear each other's feedback. The set of all edges in *E* which are incident on vertex *v* is denoted by E_v. The AP is assumed to be fully aware of the overhearing graph of its associated nodes.

The following is an outline of the proposed scheme.

BEGIN PROCEDURE

(1) The AP notifies the nodes of the order in which feedback transmissions are to be made.

(2) The nodes respond, if at all, in the specified order, transmitting their SNRs for RBs over the entire frequency band. Each turn of SNR reporting is called a stage.

(3) For each RB every node can decode the feedback transmitted by adjacent nodes and decides whether to transmit its own feedback by comparing the quantiles of its SNRs with those previously overheard.

(4) A node transmits feedback for an RB only if its SNR's quantile is higher than those of the overheard SNRs.

END PROCEDURE

In order to estimate the potential reduction in overhead achievable by the proposed scheme, we present a lower bound on the mean overhead as follows.

Lemma 1: The expected overhead (1) under the proposed scheme is lower bounded by

$$m\left(\sum_{k=1}^{n}\frac{w_k}{k}\right).$$

Proof: See appendix.

Now suppose $w_k = 1, \forall k = 1, \ldots, n$. The bound implies that one roughly needs $m \cdot \log(n)$ bits for feedback as opposed to $m \cdot n$ for full diversity gain when $\mathcal{G}$ is a complete graph. This suggests that exploiting overheard information may lead to substantial savings in overhead when $\mathcal{G}$ is dense.

3. Problem Formulation

Unless all nodes overhear each other, the order of feedback transmissions will affect the mean overhead, which we investigate in the sequel. Note that, since we have adopted quantiles as a measure for channel quality, nodes are effectively comparing i.i.d. uniform RVs associated with the quantiles of RBs over the entire frequency band at every time slot. By symmetry it is clear that we can assume $m = 1$ for the sake of simplicity of analysis and simulation regarding determining the 'best' feedback order. Thus we assume henceforth that there is only one RB per time slot.

We formulate the problem using a probabilistic graphical model as follows. In our scheme the feedback transmission by a node will influence the overhearing neighbors' feedback transmissions, and such random influence will propagate throughout the network. The precedence relations of influences are determined by the given order of feedback transmissions. Such mechanism naturally introduces modeling by a Bayesian network (BN) [1]. A BN is defined as a directed acyclic graph (DAG) where each vertex represents an RV (or a set of RVs) associated with the event of interest. The influence of a vertex on another is signified by the direction of the arc associated with the vertices. Such influence is quantified by a conditional distribution involving RVs associated with neighboring vertices. Namely a BN consists of (i) a DAG, (ii) RVs represented by vertices and (iii) the conditional distributions associated with vertices [1]. In the following we formally introduce the BN model and its associated elements.

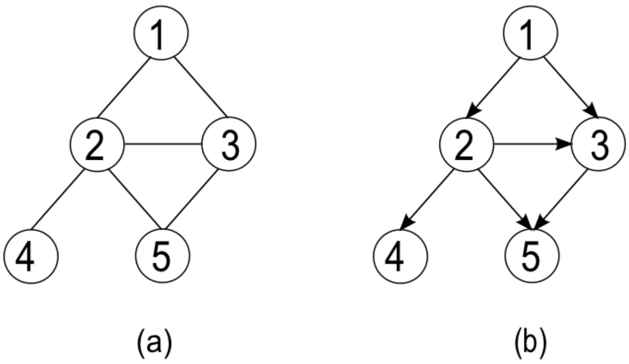

Fig. 2-(a) An overhearing graph $\mathcal{G}$.
Fig. 3-(b) A directed graph $\mathcal{G}(\pi)$ as an orientation of $\mathcal{G}$ induced by $\pi = (1,2,3,4,5)$.

Let π denote a permutation of V representing a feedback order, and $\pi(i)$ denote the i-th element of π, i.e., the index of the node associated with the i-th feedback stage. For a given undirected graph $\mathcal{G}$, a directed graph $\mathcal{D}$ is said to be an *orientation* of $\mathcal{G}$ if it is obtained by assigning certain directions to the edges of $\mathcal{G}$. Let us denote a directed arc from u to v by $\langle u, v\rangle$. By $\mathcal{G}(\pi)$ we denote the orientation of $\mathcal{G}$ such that $\langle u, v\rangle$ implies that u transmits feedback earlier than v according to π. An example of the construction of $\mathcal{G}(\pi)$ based on the order of feedback π is illustrated in **Fig. 2** where $\pi = (1,2,3,4,5)$. **Fig. 2-(b)** shows how the relations of causal influence are represented by a DAG.

The set of arcs induced by π is denoted by $E(\pi)$. Clearly $\mathcal{G}(\pi)$ is a directed acyclic graph (DAG) due to the strict ordering nature of π. For some $i \in V$ consider a pair of RVs (I_i, Γ_i) where I_i is an RV indicating whether node i will transmit feedback, and Γ_i is the SNR of the RB. Let us denote the set of parents of i in $\mathcal{G}(\pi)$ by $\Pi(i)$. We can write

$$I_i = \begin{cases} 1, & \Pi(i) = \emptyset, \\ 1\left(F_{\Gamma_i}\{\Gamma_i\} > \max_{k\in\Pi(i)}\left[F_{\Gamma_k}(\Gamma_k)\, I_k\right]\right), & \text{otherwise} \end{cases}$$

where $F_{\Gamma_i}(\cdot)$ denotes the distribution of the SNR of the RB for node *i*. We define $G_k := F_{\Gamma_k}(\Gamma_k)\, I_k$. Due to the independence of Γ_i's, clearly I_i depends only on G_k for $k \in \Pi(i)$. We construct a BN with the above setup as follows: (i) the DAG is given by $\mathcal{G}(\pi)$, (ii) we associate the vertex $i \in V$ with the RV G_i, (iii) the conditional density involving G_i is given as follows. Consider a vector $(g_1, g_2, \ldots, g_n) \in [0,1]^n$. We will denote the joint event of $\{G_k = g_k | k \in \Pi(i)\}$ by $\mathrm{Pa}(g_i)$. With slight abuse of notation we will denote the density of G_i at g_i conditional on $\mathrm{Pa}(g_i)$ by $f_{G_i|\mathrm{Pa}(G_i)}(g_i|\mathrm{Pa}(g_i))$. When $\Pi(i) = \emptyset$, we have that $I_i \equiv 1$ and $G_i = F_{\Gamma_i}(\Gamma_i)$, thus by definition $f_{G_i|\mathrm{Pa}(G_i)}(g_i|\mathrm{Pa}(g_i)) = g_i$. Since G_i depends only on $\{G_k | k \in \Pi(i)\}$ the joint distribution of $G_1, G_2, \ldots, G_n$ is given by

$$f_{G_1,G_2,\ldots,G_n}(g_1, g_2, \ldots, g_n) = \prod_{i=1}^{n} f_{G_i|\mathrm{Pa}(G_i)}(g_i|\mathrm{Pa}(g_i))$$

From the uniform distribution of $F_{\Gamma_i}(\Gamma_i)$ and the definition of G_i we have the following conditional distribution:

$$
\begin{aligned}
P(G_i \leq g_i|\mathrm{Pa}(g_i)) &= P(G_i \leq g_i, F_{\Gamma_i}(\Gamma_i) \leq \max_{k\in\Pi(i)} \{g_k\} \,|\mathrm{Pa}(g_i)) \\
&\quad + P(G_i \leq g_i, F_{\Gamma_i}(\Gamma_i) > \max_{k\in\Pi(i)} \{g_k\} \,|\mathrm{Pa}(g_i)) \\
&= P(F_{\Gamma_i}(\Gamma_i) \leq \max_{k\in\Pi(i)} \{g_k\} \,|\mathrm{Pa}(g_i)) \\
&\quad + 1(g_i > \max_{k\in\Pi(i)} \{g_k\}) P(\max_{k\in\Pi(i)} \{g_k\} < F_{\Gamma_i}(\Gamma_i) \leq g_i) \\
&= \max_{k\in\Pi(i)} \{g_k\} + 1(g_i > \max_{k\in\Pi(i)} \{g_k\})(g_i - \max_{k\in\Pi(i)} \{g_k\}).
\end{aligned}
$$

Thus we have the conditional density

$$
f_{G_i|\mathrm{Pa}(G_i)}(g_i|\mathrm{Pa}(g_i)) = \max_{k\in\Pi(i)} \{g_k\} \,\delta(g_i) + 1(g_i > \max_{k\in\Pi(i)} \{g_k\})
$$

where $\delta(x)$ is the Dirac delta function, i.e., G_i is a mixture of continuous and discrete RV. Namely $F_{\Gamma_i}(\cdot)$ is assumed to be continuous, however the probability of suppression at node i, i.e., $P(I_i = 0) = P(G_i = 0)$ can be positive, and also note that $P(I_i = 1) = P(G_i > 0)$. This completes the construction of BN model.

With slight abuse of notation let us denote the RV indicating whether node i will transmit feedback for a given directed graph $\mathcal{D}$ by $I_i(\mathcal{D})$. We define the problem of determining the optimal feedback order that minimizes (1) as follows.

Problem 1: Define $c(\cdot)$ as the cost function that maps a directed graph to average overhead, i.e.,

$$
c\left(\mathcal{D}\right) := \sum_{j\in V} w_j P(I_j(\mathcal{D}) = 1) \tag{2}
$$

for a given directed graph $\mathcal{D}$. Denote the set of all possible permutations of n nodes by $\mathcal{P}$. Our objective is to find $\pi^* = \mathrm{argmax}_{\pi\in\mathcal{P}}\, c\left(\mathcal{G}(\pi)\right)$.

We show that Problem 1 for general overhearing graphs is NP-hard as follows. Computing $c(\mathcal{G}(\pi))$ corresponds to a *marginalization* of the joint distribution $f_{G_1,G_2,\ldots,G_n}(g_1, g_2, \ldots, g_n)$ of $G_1, G_2, \ldots, G_n$. Such marginalization involving BNs associated with continuous RVs is known to be NP-hard [11], thus finding π^* is also NP-hard.

4. Optimization and Approximations

The difficulties associated with Problem 1 are twofold: (i) given a DAG computing the cost given by (2), and (ii) determining the optimal permutation π^*. To deal with those we will identify a class of DAGs of which the cost can be efficiently computed. We will then introduce so-called virtual suppression graphs (VSG) which belong to such a class. VSGs serve as a means to find a good permutation, and are intended for a simple estimation of the average overhead as well, which we discuss in the sequel.

4.1 Networks allowing efficient computation of the cost

Given a feedback order π, we will show that the cost of the associated DAG can be computed in polynomial time if the DAG satisfies the following condition.

Condition 4.1: Suppose $\mathcal{G}(\pi) = (V, E(\pi))$ satisfies the following for every node $i \in V$.

(1) For any two nodes in $\Pi(i)$, there are no common ancestors in $V \backslash \Pi(i)$.

(2) There are no other directed paths between any two nodes in $\Pi(i)$ except through the arcs in the subgraph induced by $\Pi(i)$.

Let us consider an important special case of Condition 4.1 where a DAG is a polytree or singly connected: it is easy to show that a polytree satisfies Condition 4.1. In the following we show that the cost of such DAGs can be iteratively computed, i.e., one can iteratively evaluate the probability of feedback transmission for every node. We also show that the computation involves polynomial multiplications which can be efficiently carried out, e.g., by using FFT.

Lemma 2: Without loss of generality assume $\pi = (1, 2, \ldots, n)$. Suppose the DAG associated with the BN induced by a given transmission sequence π is singly connected. Then the probability of transmission for node i can be iteratively computed as follows:

$$P(I_i = 1) = -\prod_{j \in \Pi(i)} P(I_j = 0) + \int_0^1 \left[\prod_{k \in \Pi(j)} F_{G_k}(t) \right] dt \tag{3}$$

with the distribution function $F_{G_i}(x) = P(G_i \leq x)$ given by

$$F_{G_i}(x) = P(I_i = 0) + \int_0^x \left[\prod_{k \in \Pi(j)} F_{G_k}(t) \right] dt \tag{4}$$

where $F_{G_1}(x) = x$ and $P(I_1 = 1) = 1$. Further the integrands in (3) and (4) are products of polynomial functions with rational coefficients.

Proof: See appendix. ■

By using Lemma 1 and certain properties of the graphs satisfying Condition 4.1, we show that the cost of DAGs satisfying Condition 4.1 can be computed in polynomial time as follows.

Theorem 1: The computation of the cost of $\mathcal{G}(\pi)$ satisfying Condition 4.1 has the worst case complexity of $O(d^2 n^2 \log(nd) \log(d))$ where d is the maximum degree of the underlying undirected graph $\mathcal{G}$.

Proof: See appendix. ■

In the proof of Theorem 1, we provide a computation method for the cost.

Note that Condition 4.1 simply states certain properties of a BN having an efficient computational structure, specifically when the BN represents a sequence of events which are related by the maximum of certain RVs, which is shown in the proof of Theorem 1. Condition 4.1 also characterizes DAG structures which are possibly 'dense' and multiply connected, thus the computation of the cost is seemingly difficult, however can be completed in polynomial time. In the following we leverage DAGs possessing these properties so as to find a good permutation.

4.2 A Heuristic to Determine a Good Permutation

Our objective is to construct a permutation π which represents the feedback order. We start from an empty permutation and successively add a node to π at each step. At each step we also incrementally build a DAG called *virtual suppression graph* (VSG) denoted by $\hat{\mathcal{D}} = (V, \hat{E})$ as follows. Once a node v is selected to be added to π, we convert certain edges from E_v into directed arcs which are then added to $\hat{E}$ where initially $\hat{E} = \emptyset$. Note that if we consider the cost of $\hat{\mathcal{D}}$, or $c(\hat{\mathcal{D}})$ defined by (2), one can always decrease $c(\hat{\mathcal{D}})$ by adding arcs to

$\hat{E}$. Our approach is that, we greedily select v such that adding arcs associated with v to $\hat{E}$ maximizes (with random tie-breaking) such decrease in the cost of $\hat{\mathcal{D}}$.

When assessing a candidate node v, we do not orient all edges in E_v, otherwise it may be difficult to compute the cost of the resulting $\hat{\mathcal{D}}$, especially when $\mathcal{G}$ is dense. Meanwhile we would like to orient as many edges as possible in order for $\hat{\mathcal{D}}$ to better capture the overhearing relations given by $\mathcal{G}$. To this end we propose that, for a candidate v and every edge $(v,u) \in E_v$, the arc $\langle v,u \rangle$ be added to $\hat{E}$ only if $(V, \hat{E} \cup \{\langle v,u \rangle\})$ meets Condition 4.1, otherwise (v,u) is excluded from further consideration. Thus $\hat{\mathcal{D}}$ will be an orientation of a spanning subgraph of $\mathcal{G}$ satisfying Condition 4.1. Once π is finalized, however, the actual feedback suppression will be based on $(V, E(\pi))$, i.e., overhearing opportunities are fully exploited. The cost associated with a VSG serves as an estimate on how good a feedback order is. Denote the DAG after adding arcs associated with the candidate v by $\mathcal{D}_v$. The decrease in the cost from $\hat{\mathcal{D}}$ to $\mathcal{D}_v$ can be computed as:

$$c(\hat{\mathcal{D}}) - c(\mathcal{D}_v) = \sum_{j \in \Lambda(v, \mathcal{D}_v)} w_j \left[P\left(I_j(\hat{\mathcal{D}}) = 1 \right) - P\left(I_j(\mathcal{D}_v) = 1 \right) \right]$$

where $\Lambda(v, \mathcal{D}_v)$ denotes the children of v under $\mathcal{D}_v$. This is easily seen since the difference between $\hat{\mathcal{D}}$ and $\mathcal{D}_v$ is the presence of arcs emanating from v. The algorithm is outlined in **Fig. 4**. Due to the greedy nature of successively determining the feedback order, we will refer to the proposed algorithm as Greedy Quantile-Based Sequential Feedback (G-QBSF). An example illustrating the implication of G-QBSF as well as simulation results are presented in Section 6.

Given: Overhearing graph $\mathcal{G} = (V, E)$
$V' \leftarrow \emptyset,\ \pi \leftarrow \emptyset,\ \hat{E} \leftarrow \emptyset,\ \hat{\mathcal{D}} \leftarrow (V, \emptyset)$
while $|V'| < n$ **do**
  for all $v \in V \backslash V'$ **do**
    $E'(v) \leftarrow \emptyset$
    for all $e \in E(\pi \oplus v)$ outward from v **do**
      if $(V, \hat{E} \cup E'(v))$ satisfies Condition 4.1 **then**
        $E'(v) \leftarrow E'(v) \cup \{e\}$
      else
        Remove the undirected edge associated with e from E
      end if
    end for
    $\mathcal{D}_v \leftarrow (V, \hat{E} \cup E'(v))$
    $\Delta'(v) \leftarrow c(\hat{\mathcal{D}}) - c(\mathcal{D}_v)$
  end for
  $v^* \leftarrow \text{argmax}_v \{\Delta'(v)\}$
  $\pi \leftarrow \pi \oplus v^*,\ V' \leftarrow V' \cup \{v^*\},\ \hat{E} \leftarrow \hat{E} \cup E'(v^*)$
  $\hat{\mathcal{D}} \leftarrow (V, \hat{E})$
end while

Fig. 4. The description of the proposed algorithm. A binary operator $\oplus$ is defined as the operation of adding $v \in V$ to π such that $\pi \oplus v := (\pi(1), \pi(2), \ldots, \pi(|\pi|), v)$ where $|\pi|$ is the length of π.

5. Measurement Distortions

In the proposed scheme the distribution of the overheard SNR values may not match the true distribution, since high SNR values are more likely to have been transmitted due to comparison of the quantiles of neighboring nodes' SNR. Thus the distribution of overheard SNR tends to be biased, and it is necessary to undo such biases.

Suppose that for some $i \in \Pi(j)$, $F_{G_i}(x)$ is known to j. Suppose nodes make measurements only if the parent's feedback is not suppressed. To model this let us consider a random variable H_i which has the distribution function $F_{H_i}(x) = P(G_i \leq x | G_i > 0)$. Denote the RV representing the measured SNR value by $\tilde{\Gamma}_i$ and denote its distribution function by $F_{\tilde{\Gamma}_i}$.

Lemma 3: We have that

$$F_{\Gamma_i}(x) = F_{H_i}^{-1} \circ F_{\tilde{\Gamma}_i}(x). \quad (5)$$

Proof: See appendices. ■

Now suppose a node has *independently* measured (e.g., at times with intervals greater than the coherence time) node i's SNR values k times where the measurements at l^{th} time is denoted by $\tilde{\Gamma}_i^l$ for $1 \leq l \leq k$. Consider an empirical estimator $\hat{F}_{\tilde{\Gamma}_i}^k(x)$ for $F_{\tilde{\Gamma}_i}(x)$ given by

$$\hat{F}_{\tilde{\Gamma}_i}^k(x) = \frac{1}{k} \sum_{l=1}^{k} 1\left(\tilde{\Gamma}_i^l \leq x\right) \quad (6)$$

From (5) it is clear that we would like to use the following simple estimator for the distribution of Γ_i:

$$\hat{F}_{\Gamma_i}^k(x) = F_{H_i}^{-1}\left(\hat{F}_{\tilde{\Gamma}_i}^k(x)\right). \quad (7)$$

By using the estimator (7) we mean that, when we overhear an SNR sample from i, we first compute the empirical quantile given by (6), and then we apply $F_{H_i}^{-1}(\cdot)$ to it so as to estimate the true quantile. From $F_{G_i}(x)$ it is easily seen that $F_{H_i}(x)$ is a smooth polynomial function with an inverse, thus the second step involves solving a polynomial equation, which can be easily carried out using, e.g., Newton's method. From the smoothness of $F_{H_i}^{-1}(\cdot)$ the functional law of large numbers dictates that (7) will converge pointwise to $F_{\Gamma_i}(x)$ as *k* tends to infinity [12].

Suppose the network satisfies Condition 4.1: the AP can compute $F_{G_i}(x)$ as stated in the proof of Theorem 1. Since $F_{G_i}(x)$ are polynomial functions and depend only on the network topology, the orders and coefficients of the polynomials $F_{G_i}(x)$ can be updated to the nodes for only one time under fixed topology. Note when $F_{G_i}(x)$ is known, we can remove the bias as described above.

However, if the network does not meet Condition 4.1, the exact distributions of $\{G_i\}$ will be difficult to find. In that case we introduce two phases for the feedback: measurement phase and normal phase. We fix an integer $l \geq 1$ and let the measurement phase be at every lT_c. The rest of the time normal feedback is performed. At each measurement phase we let the feedback be based on the VSG $\hat{\mathcal{D}}$, i.e., each node ignore feedback from an arc that is not in $\hat{\mathcal{D}}$. In measurement phases, the feedback structure always meets Condition 4.1, thus we can use the above mentioned correction method for measurement distortions.

6. Numerical and Simulation Results

6.1 A Numerical Example

Fig. 5-(a) shows an example of an overhearing graph $\mathcal{G}$. Note that $\mathcal{G}$ contains many cycles, thus computing the cost of an orientation of $\mathcal{G}$ does not appear simple. Assume $w_k = 1$ for $k = 1, \ldots, n$. We will apply a multi-start method, specifically Greedy Randomized Adaptive Search Procedure (GRASP) [13] to our scheme compensating for 'bad' starts often made by greedy heuristics. **Fig. 5-(b)** shows the suppression graph after applying the algorithm. The resulting VSG $\hat{\mathcal{D}}$ satisfying Condition 4.1 consists of the arcs in solid line, yielding the permutation $\pi = (4, 1, 8, 5, 3, 2, 10, 9, 7, 6)$. As stated above $\hat{\mathcal{D}}$ is meant for determining π, and the actual suppressions occur through the solid and dotted arcs. We see $\hat{\mathcal{D}}$ extends to a significant portion of $\mathcal{G}$., in particular incorporates almost all the edges associated with nodes transmitting in the earlier stages. Note that such nodes are likely to contribute more to the overall cost savings when compared to those transmitting later and thus are subject to more suppression, which is the key idea behind the algorithm.

The optimal order has been found by an exhaustive search over 10! permutations. In this example the optimal order gives cost of 4.01 where our scheme yields the cost of 4.10 which is very close to the optimal cost. A random ordering yields 5.26 on average, and it costs 10 without overhearing: our scheme saves the overhead by 22% and 59% compared to random ordering and the full feedback respectively. The results from more extensive simulation are provided in the following section.

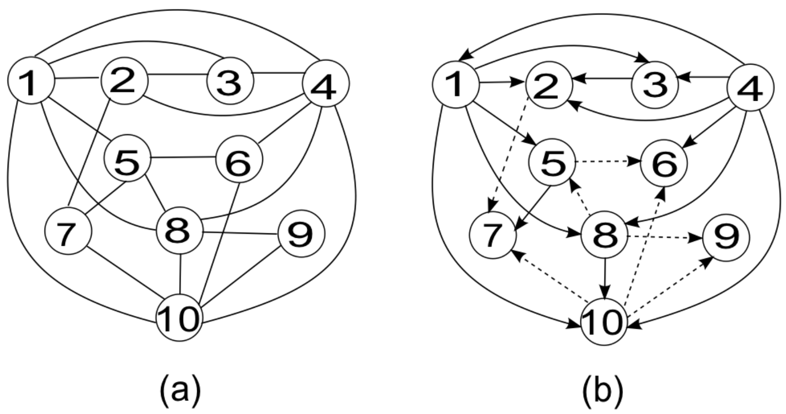

Fig. 5-(a) An example of overhearing graph $\mathcal{G}$ for $n = 10$.

Fig. 6-(b) The virtual suppression graph $\hat{\mathcal{D}}$ after applying the proposed algorithm. $\hat{\mathcal{D}}$ is an orientation of a spanning subgraph of $\mathcal{G}$ where the arcs of $\hat{\mathcal{D}}$ are represented in solid arrows.

6.2 Simulation Results

In our simulation we generated random overhearing graphs $\mathcal{G}$ such that, the adjacency between pairs of nodes was determined by i.i.d. Bernoulli RVs with probability 0.5. We associate an i.i.d. uniform RV with each node representing its quantile for the SNR of the RB. We consider two cases where uniform and nonuniform weights are assigned to users. Nonuniform weights are selected randomly from {1, 2, 3, 4} which is the set representing the number of bits required to describe the SNR of the RB of users in heterogeneous channel environments. For instance a low-end device mostly using QPSK as the modulation scheme will likely need fewer bits for feedback than high-end devices using up to 256 QAM.

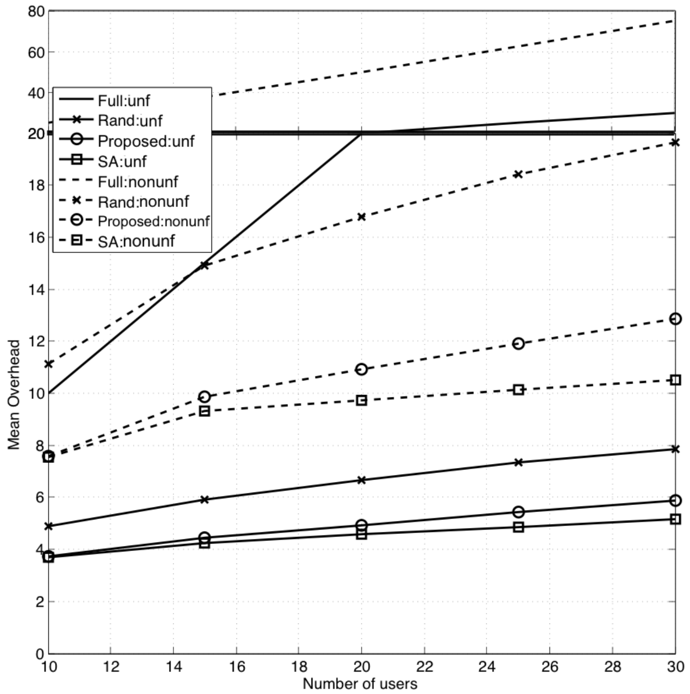

Fig. 7. Performance of the proposed scheme with uniform and non-uniform weights. Note the change on the scale in y axis from 20 and above.

1) Greedy QBSF: The plot in **Fig. 7** exhibits the performance in terms of the average overhead. Since the true optimal order is hard to find even via numerical methods for large n, we adopt an extensive search using simulated annealing (SA). We resort to the solutions obtained by SA as a reference for the 'optimal' order. The costs obtained by SA are represented by the curve labelled 'SA'. For our scheme the savings as compared to full feedback range from 63–81% and 70–83% with uniform and nonuniform weights respectively. The gain increases with n, which is similar to what we have observed from the lower bound result, i.e., the overhead grows at slower rates than that for full feedback as n increases. The savings relative to random ordering (labelled 'Rand') is on the order of 24–26% for our scheme and 24–36% for SA. With nonuniform weights the gains relative to random ordering span 33–36% and 34–48% for our scheme and SA respectively: we observe that the proposed scheme is more effective with nonuniform weights which introduce more variability into the system, and thus careful ordering of feedback becomes crucial.

2) Measurement Overhead Reduction by Branching QBSF: If $\mathcal{G}$ is dense, it can be burdensome to measure quantiles from all neighbors. To alleviate this we propose *Branching QBSF* (B-QBSF) algorithm which we describe next. A *branching* is a DAG where each node has at most one parent. We would like to find a branching whose undirected version is a subgraph of $\mathcal{G}$ so that every node to make measurements from at most one neighbor, and all other neighbors are ignored. The problem reduces to finding the minimum cost branching of $\mathcal{G}$. However unlike Chow-Liu trees [14] our objective function is not decomposable, thus the problem remains hard. Hence in B-QBSF we heuristically construct a branching which maximizes the myopic cost reduction in a similar way to the algorithm outlined in **Fig. 4**, however we simply modify Step 7 such that we check if an arc assignment to candidate node v preserves the branching property of $\hat{\mathcal{D}}$.

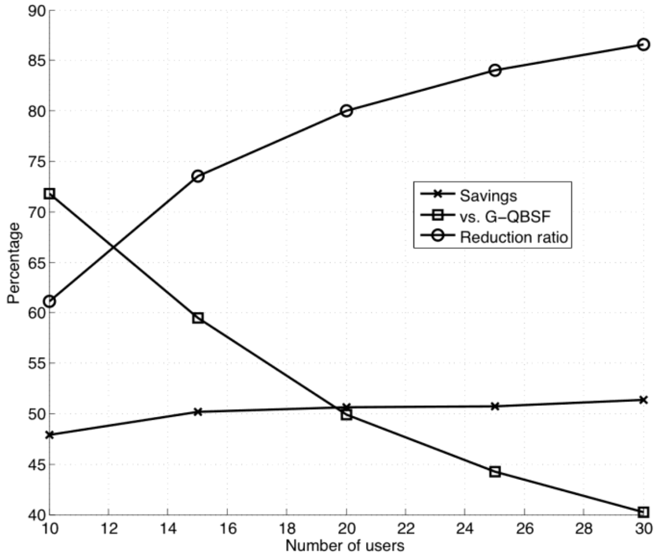

Fig. 8. Performance of B-QBSF. The curve labelled (1) ‘Savings’ represent the cost reduction achieved relative to full feedback. (2) ‘vs. G-QBSF’ represents the ratio of the overhead by G-QBSF relative to B-QBSF. (3) ‘Reduction ratio’ represents the reduction in the number of overhearing neighbors relative to G-QBSF.

The performance of the B-QBSF algorithm is shown in **Fig. 8**. B-QBSF incurs 28%-60% more overhead than G-QBSF (‘vs. G-QBSF’ curve), and this relative performance degrades with increasing n. This is partly because in our simulations the number of edges in $\mathcal{G}$ grows quadratically in n whereas the output of B-QBSF is a branching whose number of arcs grows at most linearly in n. By contrast the number of neighbors which overhear by B-QBSF is 61-87% less than that by G-QBSF (‘Reduction ratio’ curve) whereas about 50% gain relative to full feedback can be still achieved (‘Savings’ curve), which renders B-QBSF an attractive lightweight alternative to G-QBSF.

7. Conclusion

In this paper we have proposed a scheme to reduce feedback overhead in opportunistic scheduling by exploiting overhearing. We have shown that the problem of finding an optimal order for feedback can be formulated as a Bayesian network problem. We proposed a heuristic based on the conditions under which the computation of the overhead becomes tractable, which we have identified using the properties of max quantile scheduling. Future work includes extensions to data fusion in sensor networks, e.g., central station collecting maximum value among readings among a group of sensors of which the local traffic has dependency possibly from multi-hop relaying to which a BN model may be applicable.

Appendix

Proof of Lemma 1

Clearly the best case is where $\mathcal{G}$ is a complete graph. Since for each RB we compare quantiles, we are interested in the stochastic ordering of i.i.d. uniform RVs. When $\mathcal{G}$ is a complete graph, the nodes overhear all other nodes. Hence for a given RB of a neighbor at k-th stage $(2 \leq k \leq n)$, the probability that the RB will have better quantile than the previously overheard $k-1$ RBs is given by $P(U > U_{k-1})$ where the RV U_k denotes the maximum of k i.i.d. uniform RVs. Since U_k has the following probability density function

$$f_{U_k}(x) = kx^{k-1},\ x \in [0,1].$$

Thus we have that, for $k \geq 2$, $P(U > U_{k-1}) = k^{-1}$. Since we only consider the quantiles of the RBs, by symmetry the result applies to all the RBs, and the result follows.

Proof of Lemma 2

The singly connectedness of BN up to node i ensures that the observed quantiles G_j of the parents of node i are mutually independent. Also note G_j is a *mixture* of a continuous and a discrete random variable since $F_{G_i}(x)$ has a probability mass at $x = 0$, i.e., $F_{G_i}(0) = P(I_i = 0)$. Hence we have that

$$P(I_i = 1) = \int_{(0,1]} (1-t)dF_{Z_i}(t) = (1-t)F_{Z_i}(t)\Big|_0^1 + \int_0^1 F_{Z_i}(t)dt$$

where $Z_i := \max_{j\in\Pi(i)}[G_j]$. Since G_j are independent for $j \in \Pi(i)$, we have that $F_{Z_i}(x) = \prod_{j\in\Pi(i)} F_{G_j}(x)$. Thus we have that

$$\begin{aligned} P(I_i = 1) &= -F_{Z_i}(0) + \int_0^1 \left[\prod_{j\in\Pi(i)} F_{G_j}(t)\right] dt \\ &= -\prod_{j\in\Pi(i)} P(I_j = 0) + \int_0^1 \left[\prod_{j\in\Pi(i)} F_{G_j}(t)\right] dt. \end{aligned}$$

As for the distribution function $F_{G_i}(x)$ we have the following:

$$\begin{aligned} P(G_i \leq x) &= P(G_i \leq x, G_i \leq Z_i) + P(G_i \leq x, G_i > Z_i) \\ &= P(I_i = 0) + P(G_i \leq x, G_i > Z_i) \\ &= P(I_i = 0) + \int_0^1 P(G_i \leq x, G_i > z)dF_{Z_i}(z) \\ &= P(I_i = 0) + \int_0^x (x-z)dF_{Z_i}(z) \\ &= P(I_i = 0) + \int_0^x F_{Z_i}(z)dz. \end{aligned}$$

For any i the distribution function $F_{G_i}(x)$ is a polynomial function. This can be shown by induction as follows. For distribution function of the first transmitter, i.e., Node 1, it clearly holds that $F_{G_1}(x) = x$. Now suppose $F_{G_k}(x)$ is a polynomial function for some $k \geq 1$. Then from (4) $F_{G_{k+1}}(x)$ is a constant plus integration of product of polynomial functions up to stage-k nodes, thus clearly $F_{G_{k+1}}(x)$ is also a polynomial function. For example if Node 2 is a child node of Node 1, then using (4),

$$F_{G_2}(x) = \frac{1}{2} + \int_0^x F_{G_1}(t)dt = \frac{1}{2} + \frac{x^2}{2}$$

and so on.

Proof of Theorem 1

Suppose that $\mathcal{G}(\pi)$ satisfies Condition 4.1. For $i \in V$, define a set $\mathcal{E}_i$ consisting of all arcs $\langle u, v\rangle \in E(\pi)$ such that both u and v are in $\Pi(i)$. We remove arcs in $\mathcal{E}_i$ from $\mathcal{G}(\pi)$, and denote the resulting DAG by $\tilde{\mathcal{G}}_i(\pi)$, i.e., $\tilde{\mathcal{G}}_i(\pi) := (V, E(\pi) \setminus \mathcal{E}_i)$. Note that when $\mathcal{G}(\pi)$ is a polytree, $\mathcal{G}(\pi) \equiv \tilde{\mathcal{G}}_i(\pi)$ for all $i \in V$. Let us define $\tilde{G}_k^i := \Gamma_k I_k(\tilde{\mathcal{G}}_i(\pi))$. For notational simplicity let us denote the distribution function of G_i by F_i. Also denote the distribution function of $\tilde{G}_j^i$ by $\tilde{F}_j^i$.

Lemma 4: Suppose Condition 4.1 holds, then for any $i \in V$, the distribution of the maximum of $\{G_k\}$ for $k \in \Pi(i)$ is given by

$$\prod_{k \in \Pi(i)} \tilde{F}_k^i(x). \tag{8}$$

Proof: Suppose that the feedback is performed according to $\tilde{\mathcal{G}}_i(\pi)$. Without loss of generality let $\Pi(i) = \{1, \ldots, \nu\}$ where ν denotes the number of parents of Node i. In $\tilde{\mathcal{G}}_i(\pi)$, there are no directed paths between any two nodes in $\Pi(i)$ due to the removal of $\mathcal{E}_i$ and 2) of Condition 4.1. Moreover, due to 1) of Condition 4.1, any two nodes in $\Pi(i)$ do not share any ancestor. Thus the feedback transmissions from Node i's parents in $\tilde{\mathcal{G}}_i(\pi)$ will be mutually independent. In $\tilde{\mathcal{G}}_i(\pi)$, the probability of transmission at node i is determined by

$$\max\left(\tilde{G}_1^i, \tilde{G}_2^i, \tilde{G}_3^i, \ldots, \tilde{G}_\nu^i\right) \tag{9}$$

Since $\{\tilde{G}_k^i\}$, $k \in \Pi(i)$, are mutually independent, the distribution of (9) is given by (8).

Now suppose that the feedback is performed according to the original DAG $\mathcal{G}(\pi)$. We claim that the two RVs $\max(G_1, G_2, \ldots, G_\nu)$ and $\max\left(\tilde{G}_1^i, \tilde{G}_2^i, \ldots, \tilde{G}_\nu^i\right)$ are identically distributed. To see this note that the arcs in $\mathcal{E}_i$ causes feedback suppressions only among those in $\Pi(i)$, however all the transmissions, if any, which incur suppression are overheard by i. Namely $\max_{k\in\Pi(i)}\{G_k\}$ and $\max_{k\in\Pi(i)}\{\tilde{G}_k^i\}$ have the identical distribution function, which completes the proof. ■

From Lemma 2 and 4, one can compute F_i as follows.

$$F_i(x) = F_i(0) + \int_0^x \prod_{i_l \in \Pi(i)} \tilde{F}_{i_l}^i(x)dx \tag{10}$$

where i_l is the lth parent of Node i. Hence in order to evaluate $F_i(x)$ we need to compute $\tilde{F}_{i_l}^i(x)$ for $l = 1, \ldots, |\Pi(i)|$. We explain how $\tilde{F}_{i_l}^i(x)$ is computed as follows. Note Lemma 4 states that we can remove $\mathcal{E}(i_l)$ from the graph $\tilde{\mathcal{G}}_i(\pi)$ and compute the distribution of $\tilde{G}_{i_l}^i$ from the resulting graph which we denote by $\mathcal{H}$. However, $\mathcal{H}$ can be also obtained from removing $\mathcal{E}(i_l) \cup \mathcal{E}(i)$ from $\mathcal{G}(\pi)$, or equivalently, removing $\mathcal{E}_i$ from $\tilde{\mathcal{G}}_{i_l}(\pi)$. In other words, there is no causal influence from any node in $\Pi(i)$ to i_l in $\mathcal{H}$, i.e., the influences from the nodes in $\Pi(i_l) \cap \Pi(i)$ to Node i_l are absent. Hence $\tilde{F}_{i_l}^i(x)$ can be computed similar to computing F_{i_l} by considering only the influences from $\Pi(i_l) \setminus \Pi(i)$, i.e.,

$$\tilde{F}^i_{i_l}(x) = \tilde{F}^i_{i_l}(0) + \int_0^x \prod_{k \in \Pi(i_l) \setminus \Pi(i)} \tilde{F}^{i_l}_k(x)dx. \quad (11)$$

We assume that the functions $F^{i_l}_k(x)$, $k \in \Pi(i_l)$ in the integrand of (11) had been computed earlier and are available when computing (11): similarly the polynomials $\tilde{F}^i_{i_l}(x)$, $l = 1 \ldots, |\Pi(i)|$, are also stored for later use. We repeat the computation (11) for $l = 1, \ldots, |\Pi(i)|$. Then $\tilde{F}^i_{i_l}(x)$, $l = 1, \ldots, |\Pi(i)|$, are used in (10) to compute $F_i(x)$. This summarizes how we compute the cost of any DAG which satisfies Condition 4.1.

Finally we discuss the complexity of the above computation. By using induction one can easily show that, the order of $F_i(x)$ is at most i. Note we compute (11) for $|\Pi(i)|$ times which is no more than d. For each computation of (11) we multiply at most d polynomials of order i. Thus the multiplication of at most d polynomials of the order at most i is done for a total of at most (d+1) times. The complexity of computing (11) which involves the multiplication of at most d polynomials of degree at most i is at most $O(id \log(id) \log(d))$ [15]. Considering the factor $(d+1)$ at each stage, the worst case complexity is given by $O(id^2 \log(id) \log(d))$ per stage. Thus if we consider the entire stages, the overall complexity is given by $O(d^2 \log(d) \sum_{i=1}^{n} [i \log(id)]) = O(d^2 n^2 \log(d) \log(nd))$, which completes the proof.

Proof of Lemma 3

Note that H_i represents the quantile of $\tilde{\Gamma}_i$ measured in terms of the distribution function of Γ_i, i.e.,

$$\begin{aligned} P(F_{\Gamma_i}(\tilde{\Gamma}_i) \leq x) = F_{H_i}(x) &\Longrightarrow P(\tilde{\Gamma}_i \leq F^{-1}_{\Gamma_i}(x)) = F_{H_i}(x) \\ &\Longrightarrow F_{\tilde{\Gamma}_i} \circ F^{-1}_{\Gamma_i}(x) = F_{H_i}(x) \\ &\Longrightarrow F^{-1}_{\Gamma_i}(x) = F^{-1}_{\tilde{\Gamma}_i} \circ F_{H_i}(x) \end{aligned}$$

thus (5) follows.

References

[1] F. Jensen and T. Nielsen, "Bayesian Networks and Decision Graphs," *in Journal of Springer*, 2007. Article (CrossRef Link)

[2] S. Sanayei, A. Nosratinia and N. Aldhahir, "Opportunistic downlink transmission with limited feedback," in *Proc. of IEEE Transactions on Information Theory*, 2004. Article (CrossRef Link)

[3] D.J. Love, R.W. Heath Jr, V.K.N. Lau, D. Gesbert, B.D. Rao and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE Journal on Selected Areas in Communications*, vol.26, no.8, pp.1341-1365, 2008. Article (CrossRef Link)

[4] R. Agarwal, R. Vannithamby and J. M. Cioffi, "Optimal allocation of feedback bits for downlink OFDMA systems," in *Proc. of IEEE International Symposium on Information Theory*, pp. 1686–1690, Jul. 2008. Article (CrossRef Link)

[5] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks with heterogeneous users," in *Proc. International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks,* pp. 1662-1672, Mar.2004. Article (CrossRef Link)

[6] T. Tang and R. Heath, "Opportunistic feedback for downlink multiuser diversity," *IEEE Communication Letters*, vol.9, pp.948–950, Oct.2005. Article (CrossRef Link)

[7] S. Patil and G. de Veciana, “Reducing feedback for opportunistic scheduling in wireless systems,” *IEEE Trans. on Wireless Communications*, vol.6, no.12, pp.4227-4232, Dec.2007. Article (CrossRef Link)
[8] D. Park, H. Seo, H. Kwon and B. G. Lee, “A new wireless packet scheduling algorithm based on cdf of user tranmission rate,” in *Proc. IEEE Globecom*, 2003. Article (CrossRef Link)
[9] T. Bonald, “A score-based opportunistic scheduler for fading radio channels,” in *Proc. European Wireless*, 2004. Article (CrossRef Link)
[10] S. Patil and G. de Veciana, “Managing resources and quality of service in wireless systems exploiting opportunism,” *IEEE/ACM Transactions on Networking*, vol.15, pp.1046–1058, Oct.2007. Article (CrossRef Link)
[11] G. Cooper, “The computational complexity of probabilistic inference using Bayesian belief networks.,” *Artificial Intelligence*, vol.42, no.2, pp.393–405, 1990. Article (CrossRef Link)
[12] P. Billingsley, “Convergence of probability measures,” *John Wiley & Sons*, 1968. Article (CrossRef Link)
[13] T. Feo and M. Resende, “Greedy randomized adaptive search procedures,” *J. of Global Optimization*, vol.6, no.2, pp.109–133, 1995. Article (CrossRef Link)
[14] C. Chow and C. Liu, “Approximating discrete probability distributions with dependence trees,” *IEEE transactions on Information Theory*, vol.14, no.3, pp.462–467, 1968. Article (CrossRef Link)
[15] P. Burgisser, M. Clausen and M. Shokrollahi, *Algebraic complexity theory*, Springer Verlag, 1997 Article (CrossRef Link)

Seung Jun Baek received B.S. degree in electrical engineering from Seoul National University in 1998. He earned M.S. and Ph.D. degree in electrical and computer engineering from University of Texas at Austin in 2002 and 2006. Since 2009, he has been an assistant professor at Dept. of Computer and Communications in Korea University. His research interests include cellular networks, cooperative relay networks, cognitive radio and sensor networks.