Leveraging Mobile Grid Computing for Interference Alignment and Cancelation

Parul Pandey, Hariharasudhan Viswanathan, and Dario Pompili

NSF Cloud and Autonomic Computing Center
Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ
e-mail: {parul_pandey, hari_viswanathan, pompili}@cac.rutgers.edu

***Abstract*—Interference Alignment and Cancelation (IAC) aims at significantly improving the wireless channel capacity. Existing algorithms for IAC are computationally intensive, which may lead to long execution times. A practical implementation of IAC is infeasible for fast-varying channels (when the coherence time is small, e.g., less than 0.5 s). This is because a significant amount of time has to be spent on channel estimation as IAC techniques are extremely sensitive to the degree of accuracy of channel estimates, thus leaving a very small portion of time for actual data transmission. The collective computational capabilities of nodes in the neighborhood can be exploited (for parallelism) to facilitate the practical realization of compute-intensive IAC techniques. A novel resource provisioning framework, which organizes the mobile devices in the neighborhood to form an elastic resource pool – a heterogeneous mobile computing grid – is presented. This framework enables distributed execution of compute-intensive communication algorithms like IAC. The effectiveness of the approach is studied under different operational scenarios.**

***Index Terms*—Interference alignment, wireless channel estimation, distributed algorithms, mobile grid computing.**

I. Introduction

The computation and communication capabilities of mobile hand-held devices such as smart phones, tablets, netbooks, and laptops have improved tremendously due to the advances in microprocessor, storage, and wireless technologies. We propose a novel resource provisioning framework for organizing the heterogeneous computing capabilities of mobile devices in the neighborhood in order to form an *elastic resource pool* – a mobile computing grid. This local computing grid can be harnessed to enable innovative *data-* and *compute-intensive mobile applications,* which are currently not enabled due to the insufficient computing capabilities on individual mobile devices (i.e., they cannot produce meaningful results within realistic time bounds). In this paper, we explain how our proposed framework can facilitate the practical realization of compute-intensive communication optimization algorithms such as Interference Alignment and Cancelation (IAC) for Multiple Input Multiple Output (MIMO) systems.

IAC techniques, which have recently been developed for terrestrial MIMO systems [1], [2], are aimed at significantly improving the channel capacity. The problem of increasing spectrum efficiency using techniques like IAC is of great significance because the rate of improvement in radio hardware to use greater amount of spectrum resources is lower than the rate at which the wireless capacity demand increases. Hence, we need to exploit the available spectrum resources to the fullest. Most of the prior work in the area of IAC considers unrealistic assumptions such as infinite computational capabilities at the participating nodes and perfect channel knowledge [3].

Many iterative algorithms for IAC have been proposed recently [1], [4], [5], [6]. In [4], the authors present an iterative algorithm that requires exchange of information between transmitter and receiver until convergence. This algorithm depends heavily on the initial conditions used and may converge to a local minimum for certain initial conditions. Hence, the initial conditions must be chosen carefully. If the time taken for convergence of the algorithm is greater than the channel coherence time (period of time for which the channel impulse response is considered to be not varying), the results of the algorithm will no longer be useful, and will lead to poor Bit Error Rate (BER) performance. Also, the algorithm relies on perfect synchronization between transmitter and receiver. This is a very stringent condition in real-time applications. Our aforementioned mobile grid computing framework enables mobile devices to realize collaboratively this compute-intensive IAC technique (iterative algorithm) so to produce meaningful results in realistic time constraints imposed by the time-varying wireless channel.

In this paper, we investigate the effectiveness of the proposed framework for IAC via simulations under practical scenarios (realistic assumptions) such as small channel coherence times and imperfect channel knowledge. We present a study on the potential increase in the *region of feasibility* of compute-intensive IAC techniques in time- and resource-constrained scenarios. Investigation into the region of feasibility involves studying the trade-off between computational gain (in terms of speed up over stand-alone computation) versus communication overhead (and delay) incurred as well as its effect on the performance in terms of available wireless channel capacity for data transmission. This trade-off helps us define the scenarios in which the distributed realization of such IAC technique is feasible. We consider an 802.11-based mesh-network scenario – with no mobility in an indoor setting – and the mesh clients,

This work was supported by the NSF CAREER Award No. OCI-1054234.

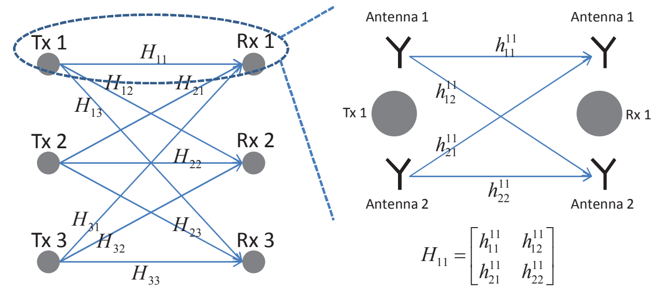


Fig. 1. An interference network with $K = 3$ users (i.e., 3 Tx-Rx pairs) with $N_T = N_R = 2$ (two antennae on each node).

e.g., laptops, mobile devices, and other wireless devices, are part of the resource pool. The following are the main contributions of this paper:

- We propose a technique to parallelize the IAC algorithm using our framework and demonstrate the gains (in terms of increase in channel capacity);
- We discuss the region of feasibility for practical realization of these algorithms using our distributed framework.

The remainder of the paper is organized as follows. In Sect. II, we present the background on IAC. In Sect. III, we propose a mobile grid computing framework and explain how we exploit it to realize distributed IAC. In Sect. IV, we discuss the performance of our proposed approach. In Sect. V, we draw conclusions and discuss how our framework can be applied to enable or to improve a broad range of compute-intensive communication applications.

II. Interference Alignment Theory

In this section, we present the necessary background on the technique of IAC and discuss a representative compute-intensive iterative algorithm for the same.

Interference Alignment: This technique provides an opportunity where each transmitter-receiver pair ("user"), in a K-user system is simultaneously able to send at a data rate equal to half of his interference-free channel capacity to his desired receiver [7]. In a K-user system, the capacity of a single-user with M antennas at both transmitter and receiver, in the absence of all interference is $Mlog(SNR) + o(log(SNR))$. In the case of IA, the network sum capacity is $\frac{KM}{2}log(SNR) + o(log(SNR))$, so the capacity per user is $\frac{M}{2}log(SNR) + o(log(SNR))$, i.e., each user gets half of the interference-free channel capacity. The pre-log factor in capacity is called Degree of Freedom (DoF), which is the number of interference-free signaling dimensions available in the system [1].

We consider a generic system model, a K-user interference channel system (shown in Fig. 1) where K transmitters are sending independent information to K receivers simultaneously so that besides a desired signal each receiver receives interference from $K-1$ links. Each transmitter T_X is equipped with N_T antennae and each receiver R_X is equipped with N_R antennae. The channel between transmitter i and receiver j is given as $\mathbf{H_{ij}}$). For the case of $N_T = N_R = 2$, $\mathbf{H_{ij}}$ is shown in Fig. 1, where each entry h_{kl}^{ij} is a complex number whose magnitude represents the signal attenuation from transmitter antenna k to receiver antenna l at each time slot (where $k, l \in 1, 2$) and whose phase represents the propagation delay. At transmitter i, $\mathbf{x_i}$ is a $d_i \times 1$ symbol vector where d_i is the number of independent information streams or the degree of freedom for the i^{th} transmitter. The goal of interference alignment is to design transmit precoding matrices $\mathbf{V_i}$ of dimensions $N_T \times d_i$ for each transmitter. The transmitted signal is then given as $\mathbf{s_i} = \mathbf{V_i x_i}$ of dimension $N_T \times 1$. These matrices are chosen such that by encoding with these matrices all the interfering signals lie in a subspace, linearly independent of the subspace of the desired signal. The heart of IA in spatial domain lies in constructing these transmit precoding vectors. The received signal vector at receiver j is given as,

$$\mathbf{r_j} = \mathbf{H_{jj} V_j x_j} + \sum_{i=1, i \neq j}^{K} \mathbf{H_{ij} V_i x_i} + \mathbf{n_j}, \tag{1}$$

where the first term is the desired signal at receiver of user j and the second term is the interference from all other users. Here, $\mathbf{n_j}$ is the $N_R \times 1$ Additive White Gaussian Noise (AWGN) or thermal noise vector.

Interference Cancelation: To cancel the interference, we have to project the received signal $\mathbf{r_j}$ on the orthogonal space of interference. The interference suppression filters ($\mathbf{U}$) of dimension $N_R \times d_i$ are used to eliminate the interference signal at the receiver and are given as $\mathbf{U_j} = null(\mathbf{H_{ij} V_i}) = null([\mathbf{H_{kj} V_k}]^{\dagger})$ (where $\dagger$ represents the Hermitian or conjugate transpose). The null space of a generic vector $\mathbf{A}$ is denoted as $null(\mathbf{A})$ and it is the set of all vectors $\mathbf{x}$ for which $\mathbf{Ax} = 0$. After applying the interference suppression filter, $\mathbf{U}^{\dagger}$, the received signal at receiver j is given as,

$$\mathbf{y_j} = \mathbf{U_j}^{\dagger} \mathbf{H_{jj} V_j x_j} + \sum_{i=1, i \neq j}^{K} \mathbf{U_j}^{\dagger} \mathbf{H_{ij} V_i x_i} + \mathbf{U_j}^{\dagger} \mathbf{n_j}. \tag{2}$$

In case of perfect IA, which is possible only when channel knowledge is perfect, the interference aligns perfectly and the interference suppression filter eliminates the interference completely. The second term in (2) is the total interference at receiver j. It is important to keep in mind that it is not possible to solve the problem optimally when channel knowledge is imperfect. Imperfect channel knowledge is mainly because of error in channel estimation or when the channel has changed since the time it was estimated. This may lead to residual interference at the receiver even after interference cancelation. A zero-forcing equalizer [8] is applied to (2) to remove the effect of channel from the received signal.

Distributed Approach for IAC: In [4], the authors present an iterative approach for IAC in MIMO systems to estimate the transmit precoding vectors and interference suppression filters. The total residual interference, called *leakage interference*, at

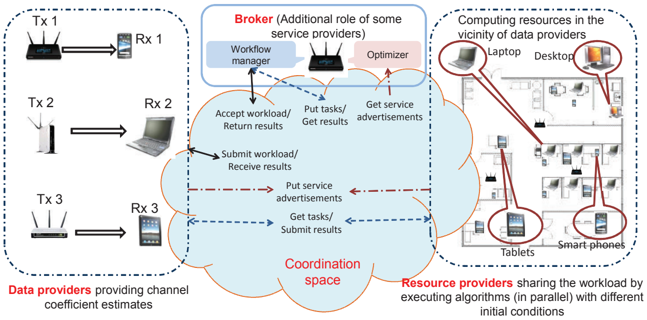


Fig. 2. An overview of the envisioned mobile computing grid for optimizing communications using compute-intensive algorithms.

the receiver of user j due to interference from all undesired transmitters ($k \neq j$) is given by,

$$\mathbf{I_j} = Tr[\mathbf{U_j}^{\dagger}\mathbf{Q_j}\mathbf{U_j}],\ \mathbf{Q_j} = \sum_{k=1,k\neq j}^{K} \frac{\mathbf{P_k}}{d_k}\mathbf{H_{kj}}\mathbf{V_k}\mathbf{V_k}^{\dagger}\mathbf{H_{kj}}^{\dagger}, \tag{3}$$

where $\mathbf{P_k}$ is the transmit power at transmitter k. Each of the d_j columns of $\mathbf{U_j}$ are given by, $\mathbf{U_{j[n]}} = \nu_n[\mathbf{Q_j}]$, $n = 1, \ldots, d_j$, where $\nu_n[\mathbf{Q_j}]$ is the eigenvector corresponding to the n_{th} smallest eigenvalue of $\mathbf{Q_j}$. In the beginning of the iterative algorithm, the transmit precoding vectors are initialized with some random values and interference suppression filters of the original network are calculated using (3) . After determining $\mathbf{U_j}$, the transmitter and receiver switch their roles. This network is called a "reciprocal" network. The estimated interference suppression filters ($\mathbf{U_j}$) of the original network now become the precoding vectors ($\overleftarrow{\mathbf{V_j}}$) for the reciprocal network. The arrow at the top indicates that this vector belongs to the reciprocal network. Similar to the original network, in the reciprocal network (with transmitters and receivers switched) the total interference leakage at receiver j due to interference from all undesired transmitters ($k \neq j$) is given by, $\overleftarrow{\mathbf{I_j}} = Tr[\overleftarrow{\mathbf{U_j}}^{\dagger}\overleftarrow{\mathbf{Q_j}}\overleftarrow{\mathbf{U_j}}]$. The interference suppression filter ($\overleftarrow{\mathbf{U_j}}$) for the receivers of the reciprocal network are calculated only to be used as the transmit precoding vectors of the original network in the next iteration. The iterative algorithm alternates between the original and reciprocal networks with only the receivers updating their interference suppression filters (in every iteration) to minimize their total leakage interference. Remember that the transmit precoding matrices ($\mathbf{V_j}$) in the reciprocal network (transmitter and receiver reversed) are the interference suppression matrices ($\mathbf{U_j}$ computed earlier) in the original network. This iteration is performed until convergence. Although the algorithm is guaranteed to converge it may not converge to global minimum due to non convex nature of interference optimization problem [4].

III. Proposed Solution

In this section, we present our framework [9] to realize compute-intensive communication applications using a mobile computing grid. The entities of the mobile computing grid may at any time play one or more of the following three logical roles: i) *service requester,* which places requests for workloads that require additional data and/or computing resources from other devices, ii) *service provider,* which can be a data provider, resource provider, or both, and iii) a *broker* (usually, the base station), which processes the requests from the requesters, determines the set of service providers that will provide or process data, and distributes the workload tasks among them. The service requester offloads (shares) the task of executing compute-intensive algorithms to (with) the service providers by submitting service requests to one of the brokers. Resource providers lend their computational (CPU cycles), storage (volatile and non-volatile memory), and communication (i.e., network interface capacity) resources for processing data. The broker is aided by a novel energy-aware resource allocation engine that will distribute the workload tasks optimally among the service providers. Note that our framework applies to applications exhibiting *data parallelism* (in which data is distributed across different parallel computing nodes that perform the same task) as well as to applications exhibiting *task parallelism* (in which parallel computing nodes may perform different tasks on the same or different data). We realize the IAC technique using data parallelism.

We consider an indoor scenario where a few pairs of nodes are communicating with each other – as shown in Fig. 2. We assume that one of the communicating nodes serves as the broker. The other wireless devices near these nodes serve as service providers and offer their computational capabilities to calculate precoding/decoding vectors with different initial conditions. These initial conditions are selected randomly by the broker and each service provider is given a unique initial condition. The broker chooses equally spaced initial conditions so as to ensure good coverage of the n-dimensional search space (in case of an n-element $\mathbf{V}$ vector) and to avoid choosing final precoding/decoding vectors based on convergence to a local minimum. Once the service providers have computed the precoding vectors, they send their results to the broker, which ranks the results and selects the best precoding/decoding vector pair based on a given criterion, i.e.,: 1) *maximization of sum-capacity*, i.e., precoding vectors that maximize the total capacity of the network are chosen, or 2) *maximization of fairness*, i.e., precoding vectors that maximize fairness (calculated using Jain's index [10]) are chosen. In the following, we describe the mobile computing grid framework by explaining the responsibilities of each node in the framework and how a distributed implementation of IAC is realized using the same.

Distributed Mobile Grid Computing Framework: The communicating nodes themselves and the nodes in their neighborhood serve as service providers.

Service discovery: The broker is made aware of the availability of service providers through voluntary service advertisements from the service providers. Service advertisements include information about the current position, amount of computing (in terms of normalized CPU cycles), memory (in bytes), and communication (in bits per second) resources, the start and end times of the availability of those resources, and the amount of residual energy at each service provider. The broker will be aware of the energy consumption profile of service provider as the information about the different types of devices is known in advance.

Workload management: Each broker is composed of two components, namely, *workload manager* and *scheduler/optimizer*, where the former handles tracking of workload requests, allocation of workload tasks, and aggregation of results; and the latter identifies the number of service providers available and determines the optimal distribution of workload tasks among them. When a service requester needs additional computing resources to process the data it generates, it submits a service request to the nearest broker and also specifies the maximum duration for which it is ready to wait for a service response. The optimizer will share the workload submitted by the requester among the available service providers based on one of several possible policies.

For example, a policy may aim at minimizing the battery drain. This can be achieved through minimization of computational load on each individual service provider by exploiting parallelism while incurring a very low communication cost. Another policy may just place emphasis on response time without considering battery drain. The set of service providers and the duration for which each of their capabilities are availed are determined by considering the trade-offs between i) the energy cost for transferring the data locally from data providers to the service providers and ii) the computational cost for availing the computational capabilities of the service providers for servicing the request and for generating the final response.

IAC Realization in the Proposed Framework: All the nodes that need to transmit data $(Tx1, Tx2, Tx3)$ submit a service request to the broker, i.e., the nodes request the broker to identify and involve other nodes in the vicinity for estimating the precoding/decoding vectors based on different initial conditions. The broker determines the number of service providers to involve (and, hence, the number of initial conditions and iterations) based on information about N_T, N_R, and DoF at the different transmitting and receiving nodes.

Algorithm Profiling: The aforementioned iterative algorithm is computationally intensive as it involves eigen-vector calculations and multiple matrix multiplications. We profiled the algorithm as specified by the authors and profiled its execution time per iteration (estimation of the transmit precoding matrices and interference suppression filter) on a mobile device (MD1: 1GHz dual-core ARM processor with 1GB RAM). The execution time (on a mobile device) per iteration increases with the number of antenna elements and DoF. The time taken by mobile device MD1 for one iteration to estimate the precoding and interference suppression vectors in one iteration with $N_T = N_R = 3$ is approximately 24 ms. The coherence time of the channel, i.e., the duration for which the channel is considered stationary, is assumed to be 1 s. Signals received within a particular coherence time interval have correlation greater than 0.5. In indoor wireless networks the coherence time is of the order of few hundred milliseconds to few seconds [11]. To perform channel estimation (conventional Least Square approach) and estimation of precoding/decoding vector from the iterative algorithm (maximum 10 iterations) will take approximately 250 ms, leaving 750 ms for the reuse of these vectors. Currently, in our evaluations, we allow each service provider to estimate only one set of precoding/decoding vectors for all K users (starting from only one initial condition per user). This is because we assume that the computing power of mobile devices is similar to that of MD1's. This number can be increased further (effectively increasing the total number of initial conditions used) in future when the computing power on devices increases further.

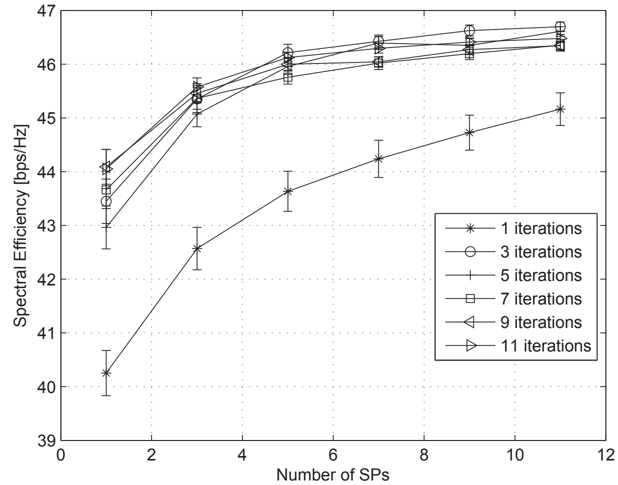


Fig. 3. Spectral efficiency vs number of Service Providers (SPs) (for different number of iterations).

Leveraging the Framework: Consider the case in which the channel coherence time T_c is 1 s. If we want to limit the fraction of time for which the IAC algorithm runs in one T_c, then the number of iterations that can be performed decreases progressively for higher-order MIMO systems (with higher number of N_T, N_R, and DoF). As this algorithm starts with an arbitrary initial condition, better performance (in terms of minimizing the residual interference) is guaranteed with a higher number of iterations, which is infeasible under fast-varying channel conditions ($T_c < 0.5$ s). However, this can be compensated for by simultaneously estimating multiple candidate precoding/interference-suppression filters and by picking the 'best' $\mathbf{V_k}, \mathbf{U_k}$ pair ($\forall\ k = 1 \dots K$).

In Fig. 3, we see that for each service provider as the number of iterations is increased the spectral efficiency also increases. This is because of the lower convergence error in the estimation of precoding/decoding vectors when the algorithm is run for higher number of iterations. As the number of iteration is increased from 1 to 11, w.r.t. one SP the spectral efficiency increases by 3 bps/Hz. For a 20 MHz-system, this translates to an increase of 60 Mbps in capacity. Considering the time taken per iteration is 24 ms, we can at maximum do 10-15 iterations for MD1 leaving the residual time for channel estimation and coordination. For a 802.11n system (64 QAM, 5/6 code rate, 2.4 GHz carrier frequency), the time taken for coordination (distribution of result among participating nodes) is 0.5 ms in case of 8 service providers. Our framework

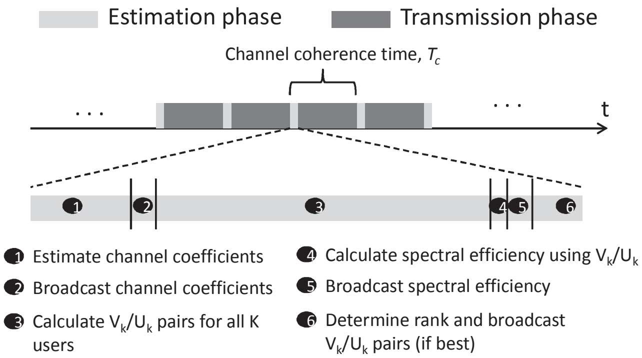


Fig. 4. An overview of our approach to distributed data-parallel IAC; sequence of events happening at every Service Provider (SP).

provides a unique opportunity to execute the algorithm over multiple initial conditions (in a data parallel manner) where each service provider uses a unique initial condition and the the best set of $\mathbf{V_k}, \mathbf{U_k}$ pair is chosen from them. In Fig. 3, as the number of service providers increases the spectral efficiency increases. This is because we get to choose those precoding/decoding vectors which maximize the spectral efficiency. In this figure as the number of SPs are increased from 1 to 11 for 1 iteration the spectral efficiency increases by 5 $\mathrm{bps/Hz}$. For a 20 MHz-system, this translates to an increase of 100 Mbps in capacity. If the coherence time of the channel is very small (say less than 0.25 s), then performing channel estimation and estimation of precoding vector from the iterative algorithm using the distributed realization of the algorithm is not possible. On the other hand, if the coherence time is of the order of multiple seconds (say greater than 3 s), each receiver or transmitter node can itself try the algorithm with different initial conditions and converge to an optimum result. Hence, the region of feasibility of our algorithm lies when the coherence time of the algorithm is of the order of few hundred milliseconds to few seconds.

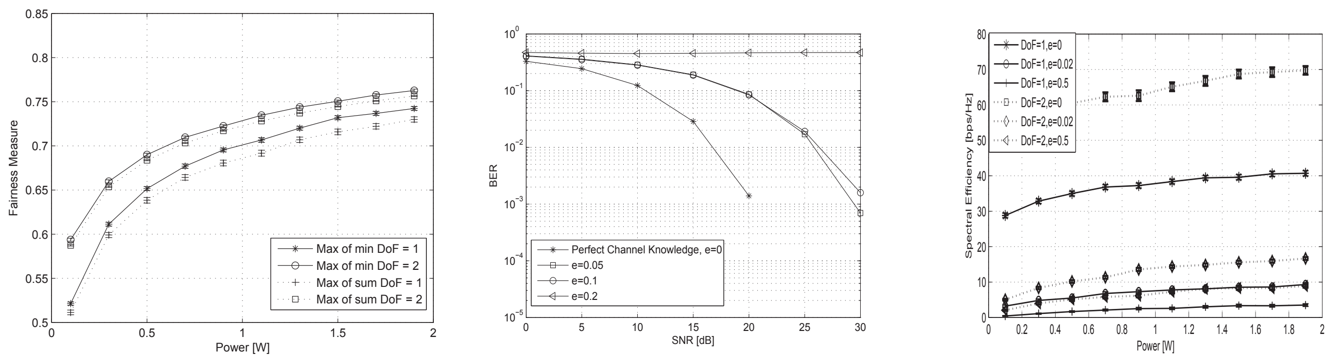
Sequence of Events at a Service Provider (SP): Fig. 4 shows the sequence of events happening at every service provider over time. We identify two distinct phases, which alternate over time: i) the *estimation phase* and ii) the *data-transmission phase*. The duration of the two phases together lasts for the channel coherence time T_c. Figure 4 also shows the sequence of events (numbered 1 through 6) within the estimation phase. Events 1 and 2 correspond to channel coefficient estimation and sharing. Event 3 corresponds to calculation of $\mathbf{V_k}$, $\mathbf{U_k}$ pairs for all K active users using the aforementioned iterative procedure. The number of iterations is determined by the fraction of the channel coherence time that we are ready to spend in the estimation of precoding/interference-suppression vectors. Here, the number of iteration is set less than 5 as it corresponds to 25% of the channel coherence time of 1 s. Events 4 and 5 correspond to the steps where the service providers collaborate to determine the best precoding/interference-suppression-vector pairs. Once the best vector pairs are determined, the corresponding transmitter and receiver nodes are provided with the vectors (corresponds to Event 6), which are then used in the actual data transmission.

IV. Performance Evaluation

We implemented the IAC algorithm using our framework and studied its performance via simulations. Our simulations are geared towards understanding the performance gain (in terms of achievable spectral efficiency). We consider a 20 MHz-channel as in 802.11n system. The modulation scheme used is 64-QAM and we used a 5/6 coding rate. Each node has $N_T = N_R = 2$ antennae and transmits only one information symbol at a time, unless specified otherwise. We model the channel gains using log-distance path loss model [12], according to which the received power (in dBm) at a distance d (in meters) from the transmitter is given by $Pr(d) = Pr_o - 10\gamma \log(d) + X_\sigma$, where Pr_o is the loss at distance of 1 m from the transmitter, X_σ represents flat fading and is modeled as a normal random variable with standard deviation σ. The path-loss exponent depends on the building and the environment type. We assume an indoor environment setting with γ set to 3 and σ set to 7 [13]. The transmit power in our simulation is varied from 0.1 to 2 W. The topology of the network in all our simulations is fixed. We consider that service providers and communicating nodes have computational capability similar to MD1. The coherence time of the channel is assumed to be 1 s and, hence, we consider a maximum of 10 iterations corresponding to MD1's computational capabilities.

Figure 5(a) shows a comparison (in terms of Jain's fairness index) of min-max and maximum sum rate technique when choosing the precoding and interference suppression filters for DoF equal to 1 and 2 as the power is varied. We can see that max-min technique gives higher fairness than max sum-rate technique. The broker can choose the precoding/decoding vectors for transmission by using one of the two techniques.

We consider that each receiver node estimates the channel from itself to all the transmitters. Each receiver then broadcasts its channel estimates to all the service providers. The error in estimation of channel coefficients can be modeled as $\bar{\mathbf{H}} = \mathbf{H} + e\mathbf{\Omega}$. We consider the channel estimation error model as in [14], where $e\mathbf{\Omega}$ is the estimation error, which is uncorrelated with $\mathbf{H}$. The entries of $\mathbf{\Omega}$ are i.i.d. zero-mean complex Gaussian with unity variance and e is the measure of how accurate the channel estimation is. We consider variation in BER with Signal to Noise Ratio (SNR) as a metric to evaluate the framework performance. Figure 5(b) shows the variation of BER as the SNR is varied. We see that BER increases as the channel estimation error e increases. The value of e is varied from 0 to 0.2, where the value of 0 indicates perfect channel knowledge. Figure 5(c) shows the variation in spectral efficiency with variation in power for different degrees of freedom. For DoF=2, as the measure of channel error (e) increases from 0 to 0.2, the spectral efficiency decreases by 70% for a power of 2 W. For DoF=1, the spectral efficiency decrease by 75%. Both the metrics, BER and Spectral Efficiency, indicate the sensitivity of IAC technique to the accuracy of channel knowledge. Hence, performance of a practical implementation of IAC depends largely on the quality of channel estimates. Figure 6 shows the variation of



(a) (b) (c)

Fig. 5. (a) Performance in terms of fairness among $K = 3$ users, $N_T = N_R = 2$; (b) BER vs SNR for different degrees of inaccuracy in channel estimates ($K = 3$, $N_T = N_R = 2$); (c) Performance in terms of spectral efficiency for different degrees of inaccuracy in channel estimates (DoF equal to 1 and 2).

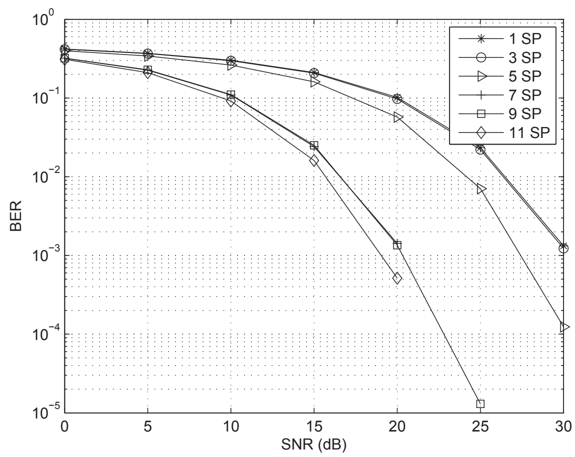


Fig. 6. BER vs SNR ($K = 3$, $N_T = N_R = 2$) with imperfect channel knowledge ($e = 0.2$).

BER with SNR as the number of SPs is increased from 1 to 11. The channel knowledge is assumed to be imperfect. The value of e in the channel model is assumed to be 0.2. We can see that as the number of service provider increases, the BER decreases. This is because of the selection of precoding vectors received from multiple service by employing multiple initial conditions. The performance of the algorithm by employing 11 SPs, as seen in Fig. 6, is very similar to the BER performance in the case of perfect channel knowledge ($e = 0$), as seen in Fig. 5(b).

V. Conclusions and Future Work

We proposed a mobile computing framework to implement Interference Alignment and Cancelation (IAC) technique, which is a compute-intensive communication realization problem. We presented the feasibility of IAC in an actual mesh static environment. We studied how the performance of our distributive realization varies as the number of service providers and algorithm iterations are varied. We observed that the efficiency of IAC largely depends on the quality of channel estimates and presented an analysis of IAC under our distributed computing framework when the channel knowledge is not perfect. We saw that under our framework the performance of IAC is better than existing IAC algorithms in case of imperfect channel knowledge. As future work, we plan to distribute joint channel estimation and estimation of precoding technique using our framework. We also plan to incorporate intelligence at the receiver (broker) to select the best set of service provider from the nodes in the neighborhood.

References

[1] V. Cadambe and S. Jafar, "Interference Alignment and Degrees of Freedom of the K-user Interference Channel," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, 2008.

[2] A. Ghasemi, A. Motahari, and A. Khandani, "Interference alignment for the k user mimo interference channel," in *Proc. of IEEE International Symposium Information Theory Proceedings (ISIT)*, 2010, pp. 360–364.

[3] X. Li and H. Fan, "Blind Channel Identification: Subspace Tracking Method Without Rank Estimation," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2372–2382, 2001.

[4] K. Gomadam, V. Cadambe, and S. Jafar, "A Distributed Numerical Approach to Interference Alignment and Applications to Wireless Interference Networks," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3309–3322, 2011.

[5] D. Papailiopoulos and A. Dimakis, "Interference alignment as a rank constrained rank minimization," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2010, pp. 1 –6.

[6] S. Jafar and S. Shamai, "Degrees of Freedom Region of the MIMO X Channel," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 151–170, 2008.

[7] S. Jafar, *Interference Alignment-A New Look at Signal Dimensions in a Communication Network*, 2011.

[8] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge Univ Pr, 2005.

[9] H. Viswanathan, E. K. Lee, I. Rodero, and D. Pompili, "An Autonomic Resource Provisioning Framework for Mobile Computing Grids," in *Proc. of IEEE International Conference on Autonomic Computing (ICAC)*, San Jose, CA, Sept. 2012.

[10] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 2008.

[11] A. Keshavarzian, E. Uysal-Biyikoglu, D. Lal, and K. Chintalapudi, "From Experience with Indoor Wireless Networks: A Link Quality Metric that Captures Channel Memory," *Communications Letters, IEEE*, vol. 11, no. 9, pp. 729–731, 2007.

[12] T. Rappaport and S. B. O. (Firme), *Wireless Communications: Principles and Practice*. Prentice Hall PTR Upper Saddle River (New Jersey), 1996, vol. 2.

[13] D. Faria, "Modeling Signal Attenuation in IEEE 802.11 Wireless LANs-vol. 1," *Computer Science Department, Stanford University*, 2005.

[14] C. Wang, E. Au, R. Murch, W. Mow, R. Cheng, and V. Lau, "On the Performance of the MIMO Zero-Forcing Receiver in the Presence of Channel Estimation Error," *IEEE Transactions on Wireless Communications*, vol. 6, no. 3, pp. 805–810, 2007.