Distributed transmit beamforming with one bit feedback revisited: how noise limits scaling

Muhammed Faruk Gencel*, Maryam Eslami Rasekh†, Upamanyu Madhow‡
Department of Electrical and Computer Engineering
University of California Santa Barbara
Santa Barbara, California 93106
Email: *gencel@umail.ucsb.edu, †merasekh@ece.ucsb.edu, ‡madhow@ece.ucsb.edu

***Abstract*—Distributed transmit beamforming with N cooperating nodes, each with fixed transmit power, provides a received power scaling with N^2, corresponding to a "power pooling" gain of N and a beamforming gain of N. Prior work has shown that the optimal beamforming solution can be attained using a decentralized, iterative algorithm based on one bit (per iteration) feedback broadcast from the receiver to the transmitters. The algorithm is provably convergent in a noiseless setting, and is the basis for several successful prototypes. In this paper, we develop a framework for providing analytical insight into the effect of receiver noise, with the following key question in mind: can we bootstrap the algorithm from the incoherent power-pooled solution to operate in a regime in which the received SNR *per node* can be made arbitrarily small as we scale up the number of nodes N? Our analytical computations, validated by simulations, yield a somewhat negative answer: while the power-pooling gain guarantees a linear increase in received power with N, the per-node SNR cannot be scaled down with N if we wish to attain a quadratic increase in received power. Specifically, the *fraction* of the ideal beamforming gain attained using the one-bit algorithm is asymptotically independent of N, and depends only on the per-node SNR. However, the one-bit algorithm provides significant performance gains in practical regimes with a moderate number of cooperating nodes: the per-node SNR required for attaining a substantial fraction of the beamforming gain is low enough (e.g., - 5dB for 65% of the beamforming gain) to provide significant extension in operation regimes, while providing aggregate SNRs which permit reliable communication at high spectral efficiency: for example, starting from -5 dB per-node SNR, we obtain about 11 dB aggregate SNR with 10 cooperating nodes, and 17 dB SNR with 20 cooperating nodes.**

I. Introduction

Distributed MIMO refers to cooperation between transmit or receive nodes to organize themselves into virtual antenna arrays. We consider here the problem of distributed transmit beamforming, in which N nodes coordinate their transmissions to send a common message to a distant destination. If the per-node transmit power does not scale down with N, then the net gain in received power from distributed beamforming scales as N^2: a power pooling gain of N because the virtual array is sending at N times the power compared to a single transmitter, and a beamforming gain of N because the signals from different transmitters combine coherently at the receiver. Feedback, whether explicit or implicit (e.g., via channel reciprocity in TDD systems), is crucial for this purpose, since open-loop beamforming is too sensitive to errors in location estimates. Furthermore, the cooperating nodes must be tightly synchronized to achieve coherence at the receiver.

Assuming that implicit feedback using channel reciprocity is not available (as in an FDD system, for example), the conventional approach for centralized MIMO is to feed back explicit channel estimates, or cleverly quantized versions thereof, for each transmit antenna, with the estimates based on pilot signals sent from each antenna. This approach does not scale well to distributed MIMO systems with a large number of nodes. An attractive alternative is decentralized phase adaptation based on *aggregate* feedback that does not depend on the individual identities of the transmitters. Specifically, an iterative, decentralized, randomized ascent algorithm based on one bit feedback (regarding changes in received signal strength per iteration) is provably convergent [1], and provides the basis for several prototypes [2], [3] by virtue of its simplicity and scalability in *protocol* terms: the receiver is oblivious of the number and identities of cooperating transmitters. Another potential advantage of the one bit algorithm is that its initial condition is the incoherent power pooling solution which already has an N-fold gain in average received power (of course, the actual power is subject to fading due to the lack of coordination of transmit phases). In this paper, we ask whether these advantages can be translated into "indefinite" scalability in *physical* terms: given the N^2 scaling of received power with ideal beamforming, can we reduce the transmit power per node or increase the range indefinitely, simply by increasing N? In this paper, we show that, when we take into account noise at the receiver, the answer is no, at least for the one-bit algorithm in its present form. Specifically, if we wish to attain a certain fraction of the ideal beamforming gain (on top of the power pooling gain), then the per-node SNR must be above a threshold which does not scale down with N. Fortunately, however, such thresholds are still low enough that significant performance gains are obtained in practical regimes of tens of cooperating nodes.

Approach and contributions: Our starting point is the simple model and analytical framework developed in [1]. In the one bit algorithm, each transmitter randomly perturbs its phase at each iteration, and the receiver broadcasts one bit of feedback regarding whether the received signal strength (RSS) has gotten better or worse. If better, then the transmitters keep their perturbations, and if worse, they undo them. We now assume that the received samples are corrupted by complex WGN, so that the feedback is regarding changes in the magnitudes of the noisy received samples. This requires a modification of the feedback generation mechanism, with the receiver performing noisy RSS comparisons over a window to avoid deadlocks due to noise. The process is modeled as a Markov chain, with transition probabilities computed using joint Gaussian approximations on random variables associated with the RSS and its one-step evolution. Drift analysis on the expected RSS based on this framework yields the fraction of the ideal beamforming gain that the algorithm converges to. Numerical results from these analytical approximations (based on asymptotics in N) match closely with simulations for even moderately large values of N.

We observe from our analysis that the converged beamforming gain monotonically increases with N, the *fraction* of the N^2-fold gain in received power that is obtained depends on per-user SNR alone and is independent of N. Thus, the increased initial power pooling gain at higher N does not help overcome receiver noise. In essence, this is because the expected RSS increase in each step due to randomized phase perturbations (whose optimal variance scales as $\frac{1}{N}$) exhibits constant scaling (independent of N), and is therefore comparable to the receiver noise variance. Thus, the effect of noise cannot be countered by increasing N.

Related Work: While our interest here is in noise at the receiver, the analysis in this paper is similar to recent work on analyzing the impact of phase noise on distributed receive beamforming [4], both building on the original noiseless analysis in [1]. The problem of synchronization for distributed MIMO applications has been the subject of many studies. In WiFi-like systems, TDD-based reciprocity plus coordination across wired backhauls has been employed for access point cooperation [5]–[7]. Distributed MIMO in a cellular context [8] makes heavy use of wired backhaul and explicit feedback. There are a number of all-wireless DMIMO systems that do not rely on reciprocity, but use some version of the one-bit feedback algorithm [3], [9], [10] or per-node feedback [11].

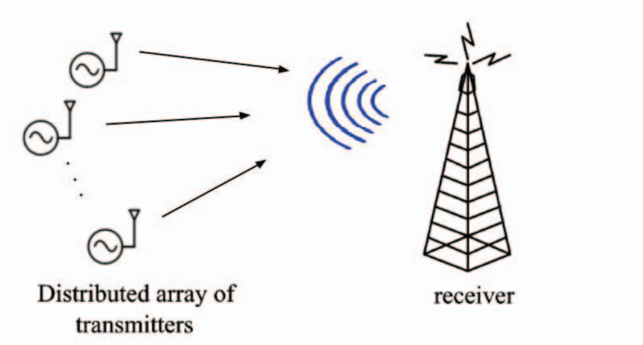

Fig. 1. System model of N transmitters beamforming with feedback broadcast from the receiver.

II. System Model

We consider the simplest possible system model to study the effect of noise on the one bit algorithm. As depicted in Figure 1, N transmitters send a common message to a receiver over a noisy, flat fading channel, with feedback from the receiver used by the transmitters to adjust their phases so as to align at the receiver. We assume that the transmitters are synchronized in timing and frequency. Ignoring the common message, the received sample is given by

$$Z_n = \underbrace{\sum_{i=1}^{N} a_i e^{j(\theta_i+\gamma_i+\psi_i)}}_{:=Y} + \bar{w}, \quad (1)$$

where $h_i = a_i e^{\psi_i}$ is the complex gain of the channel seen by transmitter i,γ_i is the receiver's phase offset relative to transmitter i, θ_i is the adjustable phase for transmitter i, and $\bar{w} \sim CN(0, N_0)$ is the receiver noise. *The subscript "n" refers to the noise corrupting the received sample, and hence the RSS estimate.* The received signal strength (RSS) is given by $Y_n = |Z_n|$. The ideal beamforming solution corresponds to $\theta_i = -(\gamma_i + \psi_i)$ (modulo any constant phase offset independent of i), and yields the maximum possible value of noiseless RSS as $Y_{max} = \sum_{i=1}^{N} a_i$.

It is convenient to express phases taking as reference the direction of the received sample. Setting $\phi_i = \theta_i + \gamma_i + \psi_i - \angle(Z_n)$, we can write

$$Y_n = \left| \sum_{i=1}^{N} a_i e^{j\phi_i} + w \right|, \quad (2)$$

where $w \sim CN(0, N_0)$: the statistics of the noise are unchanged under the change of phase reference due to circular symmetry.

III. The one bit algorithm

The one bit algorithm is described as follows. The feedback $F(k)$ broadcast by the receiver at the end of time slot k is generated by comparing the current RSS with the best RSS from among the last M iterations, as follows.

$$F(k) = \begin{cases} 1 & Y_n(k) > Y_{n,best}(k) \\ 0 & Y_n(k) < Y_{n,best}(k) \end{cases}, \quad (3)$$

where

$$Y_{n,best}(k) = \max_{k-M \le t < k} Y_n(t)$$

Each transmitter updates its phase according to the feedback from receiver as follows.

$$\theta_i(k+1) = \begin{cases} \theta_i(k) & F(k) = 0 \\ \theta_i(k) + \delta_i & F(k) = 1 \end{cases}. \quad (4)$$

The one bit algorithm originally analyzed in a noiseless setting in [1] considers $M = \infty$ (i.e., the current RSS is compared against the best RSS seen so far). This can be problematic in our noisy setting if the receiver observes a noisy RSS value that is higher than the noiseless RSS

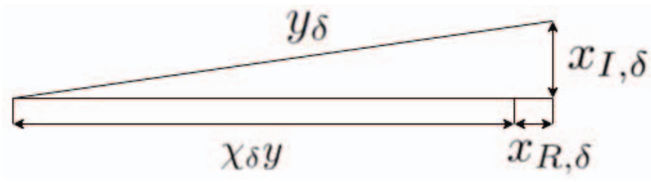

Fig. 2. Effect of random phase perturbations on the total received signal

from the current set of beamforming phases alone. This can cause the algorithm to get stuck, with actual improvements in beamformed RSS going unnoticed unless the improvement is higher than the deviation due to noise. We therefore require a windowing mechanism to enable forgetting such outliers due to noise.

A. Performance Analysis

Our goal is to characterize the progression of the RSS in a noisy setting, and to estimate the value at which it saturates. We do this by estimating the expected change, or drift, in RSS over a single iteration, and finding the point at which this drift becomes zero. We model algorithm dynamics associated with windowing as a Markov chain, depicted in Fig. 3 and discussed in more detail later, with the state at time t equal to S_i if the maximum RSS over the current window was observed at time $t-i$. The transition probabilities for this Markov chain model depend on the statistics of the noisy and noiseless RSS before and after phase perturbations over one step of the algorithm, and are characterized using jointly Gaussian approximations (which are found to be accurate even for moderately large N).

Consider the noisy and noiseless RSS values at a given iteration k (suppressed from the notation). Assuming channel gains $a_i = 1$ for simplicity, define the noiseless and noisy normalized RSS values (normalized by the maximum value of N) before phase perturbation as

$$y = \frac{1}{N}Y = \frac{1}{N}|\sum_{i=1}^{N} e^{j\phi_i}|$$

$$y_n = \frac{1}{N}Y_n = \frac{1}{N}|\sum_{i=1}^{N} e^{j\phi_i} + w_1|$$

The corresponding noiseless and noisy normalized RSS values after random phase perturbations are applied at transmitters are given by

$$\begin{aligned} y_\delta &= \frac{1}{N}|\sum_{i=1}^{N} e^{j(\phi_i+\delta_i)}| \\ y_{\delta n} &= \frac{1}{N}|\sum_{i=1}^{N} e^{j(\phi_i+\delta_i)} + w_2|, \end{aligned} \tag{5}$$

where w_1, w_2 are i.i.d. $CN(0, N_0)$, and $\{\delta_i\}$ are i.i.d. (distribution specified later).

Following the approach in [1], conditioned on the normalized noiseless RSS y prior to phase perturbation, the evolution of normalized RSS is illustrated in Fig. 2. For large N, application of the central limit theorem allows us to model the real and imaginary parts of the increments as independent Gaussian. Furthermore, the imaginary part of the increment (orthogonal to the current direction of the received sample) can be neglected, yielding the approximation [1]

$$y_\delta \approx \chi_\delta y + x_{R,\delta} \tag{6}$$

where $\chi_\delta = \mathrm{E}[cos\delta_i]$ and $x_{R,\delta} \sim N(0, \sigma_\delta^2)$ with variance

$$\sigma_\delta^2 = \frac{1 - \chi_\delta^2 - \rho_\delta \kappa(y)}{2N} \tag{7}$$

where

$$\rho_\delta = \chi_\delta^2 - \mathrm{E}[cos(2\delta_i)], \quad \kappa(y) = \frac{I_2(m)}{I_0(m)}$$

and m is derived from

$$y = \frac{I_1(m)}{I_0(m)}$$

where I is the modified Bessel function of the first kind.

It now becomes possible to approximate the noisy RSS simply by adding the real part of the noise sample to the noiseless RSS:

$$\begin{aligned} y_n &\approx y + w_{R,1} \\ y_{\delta n} &\approx \chi_\delta y + x_{R,\delta} + w_{R,2} \end{aligned} \tag{8}$$

where $w_{R,1}$ $w_{R,2}$ are the real parts of w_1/N and w_2/N, and are therefore modeled as i.i.d. $N(0, \sigma_n^2)$, with variance $\sigma_n^2 = \frac{N_0}{2N^2}$.

The receiver knows the one-step change in the noisy RSS, given by $U \triangleq y_{\delta n} - y_n$, whereas we would like to make decisions based on the one-step change in the noiseless RSS, given by $V \triangleq y_\delta - y$. Under our approximations (6) and (8), U and V are jointly Gaussian:

$$\begin{bmatrix} U \\ V \end{bmatrix} \sim N\left(\begin{bmatrix} (\chi_\delta - 1)y \\ (\chi_\delta - 1)y \end{bmatrix} \quad \begin{bmatrix} \sigma_\delta^2 + 2\sigma_n^2 & \sigma_\delta^2 \\ \sigma_\delta^2 & \sigma_\delta^2 \end{bmatrix}\right). \tag{9}$$

As described shortly, we use such joint Gaussian approximations to compute the transition probabilities for the Markov chain modeling the algorithm dynamics.

Note that, in [1] it has been shown that the optimal choice for δ_i is a distribution with variance scaling as $1/N$. Thus, we choose phase rotations from uniform distribution $\delta_i \sim U(-c/\sqrt{N}, c/\sqrt{N}]$, where c is a constant chosen based on simulation. This is easily seen to imply that $\sigma_\delta^2 \sim \frac{1}{N^2}$, which is the same scaling as the noise variance σ_n^2. Thus the entries of the covariance matrix in (9) scale with $1/N^2$, with the relative strengths of the phase perturbation and noise terms being independent of N.

B. Performance Analysis with RSS Memory

In order to compute expected value of the RSS drift given noiseless RSS value y, we make the simplifying assumption that RSS drift at iteration k is statistically independent of the feedback before $k - M$.The RSS drift at the kth step

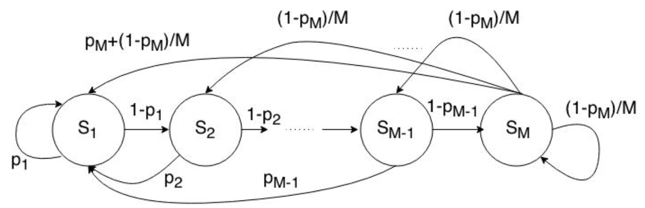

Fig. 3. Markov chain

conditioned on current state S_m and noiseless RSS value y with memory size M can be written as

$$\text{Drift}(RSS_k|S_m, y) = \mathbb{E}[V_k| \text{ feedback since } k-m, y].$$

The total RSS drift with memory size M can be computed as

$$\text{Drift}(RSS_k|y) = \sum_{m=1}^{M} \mathbb{P}(S_m|y) \cdot \text{Drift}(RSS_k|S_m, y). \quad (10)$$

The conditional drift in (10) can be expressed as:

$$\begin{aligned}&\text{Drift}(RSS_k|S_m, y) = \\ &\quad \mathbb{P}(U_k > 0|S_m)\text{Drift}(RSS_k|S_m, U_k > 0, y) \quad (11)\\ &\quad +\mathbb{P}(U_k < 0|S_m)\text{Drift}(RSS_k|S_m, U_k < 0, y).\end{aligned}$$

We model the progress of the algorithm with the Markov chain shown in Fig. 3. A positive feedback causes a transition to S_1 from any state. Negative feedback causes a transition from S_i to S_{i+1}, except when the final state is S_M. When we get negative feedback in state S_M, we assume that we transition to any of the M states with equal probability. The state transition probability matrix can therefore be written as

$$\mathbf{P} = \begin{bmatrix} p_1 & 1-p_1 & 0 & \cdots & 0 \\ p_2 & 0 & 1-p_2 & \cdots & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots \\ p_{M-1} & 0 & \ddots & 0 & 1-p_{M-1} \\ p_M + \frac{1-p_M}{M} & \frac{1-p_M}{M} & \cdots & \frac{1-p_M}{M} & \frac{1-p_M}{M} \end{bmatrix}$$

The state transition probabilities p_m are the probabilities of having positive feedback when the algorithm is in state S_m and can be defined as

$$\begin{aligned}p_m =&\mathbb{P}(U_k > 0|S_m, y) \\ =&\mathbb{P}(U_k > 0|U_{k-1} < 0, ..., U_{k-m+1} < 0, y)\end{aligned} \quad (12)$$

which can be computed from multivariate Gaussian random vector $[U_k, U_{k-1}, ..., U_{k-m+1}]^T$.

We define the probability of the algorithm being in state m conditioned on noiseless RSS value y as $\mathbb{P}(S_m|y)$. The steady state stationary distribution of the Markov chain is computed as the left eigenvector of $\mathbf{P}$ corresponding to eigenvalue of 1.

The positive feedback drift term in equation (11) is computed as

$$\begin{aligned}&\text{Drift}(RSS_k|S_m, U_k > 0, y)) \\ &=\mathbb{E}(V_k|U_k > 0, U_{k-1} < 0, ..., U_{k-m+1} < 0, y).\end{aligned} \quad (13)$$

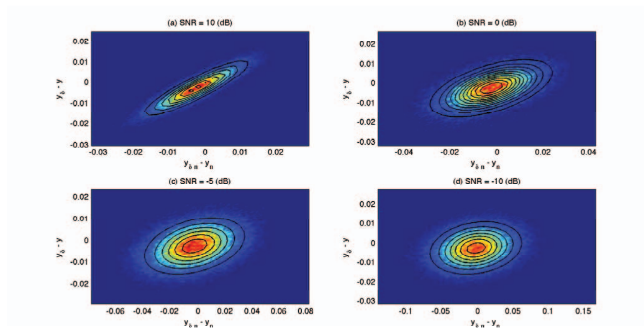

Fig. 4. 2D histogram of noiseless versus noisy RSS increments ($N = 100$ transmitters and window size $M = 4$) compared against the analytical joint Gaussian distribution.

where we compute this expectation from the multivariate Gaussian distribution $[V_k, U_k, U_{k-1}, ..., U_{k-m+1}]$.

The negative feedback drift term in equation (11) is 0 except when the algorithm is in state M since phase rotations δ_i are discarded as receiver broadcasts a negative feedback. When the algorithm is in state M and negative feedback is received then a new RSS value is considered in the next iteration. RSS drift for state M can be expressed as

$$\begin{aligned}&\text{Drift}(RSS_k|S_M, U_k > 0, y)) = \mathbb{P}[\text{max noisy RSS at } k-r] \\ &\quad \times\mathbb{E}[V_{k-r}|U_{k-r} < 0, \{U_i < U_{k-r}\}_{,\forall i \neq (k-r), i > k-M}] \\ &\quad =\mathbb{E}(V_k|U_k < 0, U_{k-1} < U_k, ..., U_{k-M+1} < U_k, y).\end{aligned} \quad (14)$$

where we compute this expectation from multivariate Gaussian vector $[V_k, U_k, U_{k-1}, ..., U_{k-M+1}]^T$ using Monte Carlo integration.

The total expected RSS drift is computed by combining (10) and (11) where we use our computed values for state probabilities, state transition probabilities and expected drift values for a given state. We estimate the steady state value of the noiseless normalized RSS y as the value corresponding to the zero crossing point of expected RSS drift conditioned on y.

IV. Numerical Results

In this section, we present simulation results for our proposed architecture and compare them with our analytical approximations.

Fig. 4 shows that the simulated histogram of (U, V) corresponds closely to our joint Gaussian analytical approximation. As SNR decreases, the correlation of U and V decreases, and hence the probabilities of the receiver broadcasting the correct decision ($P[U > 0|V > 0, y]$ and $P[U < 0|V < 0, y]$) approach $1/2$ (if U becomes conditionally independent of V, then these approach $P[U > 0|y] = P[U < 0|y] = \frac{1}{2}$.

Analytical computations for expected drift in different noise settings are plotted in Fig. 5. We see that the algorithm's progress is faster when the normalized RSS y is smaller, and that the steady state corresponding to the zero crossing values get worse as per-node SNR degrades.

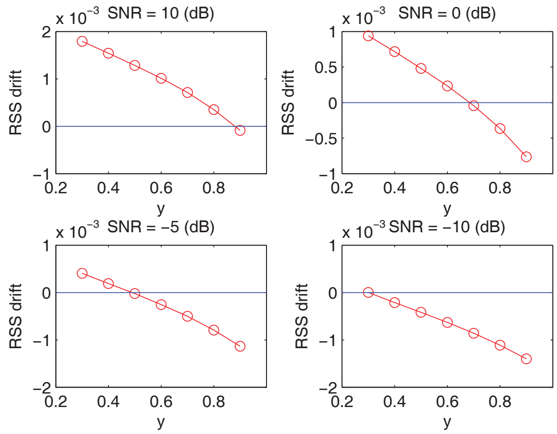

Fig. 5. Theoretical values of RSS drifts with window size $M = 4$ and for $N = 100$ transmitters in different noise levels

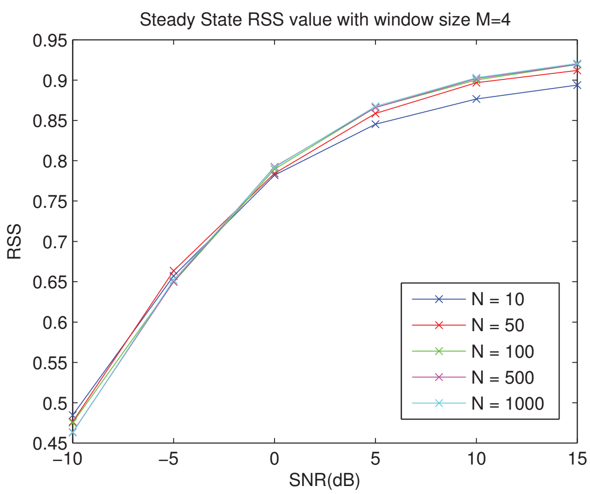

Fig. 6. Simulated normalized RSS values vs SNR for different number of transmitters, RSS memory size $M = 4$ and random phase rotations $\delta_i \sim \mathcal{U}(-\Delta, \Delta)$

Fig. 6 shows simulation results for the normalized RSS values vs per-node SNR. These are "steady state" values obtained after $100N$ iterations, averaged over multiple runs, for different number of transmitters N. Random phase rotations are generated from uniform distribution $\delta_i \sim \mathcal{U}(-\Delta, \Delta)$ where $\Delta = 100/\sqrt{N}$. We observe that the normalized steady state RSS values from simulations match with the corresponding zero crossing values of expected RSS drift in Fig. 5. The match between our analytical results and simulations does degrade slightly at lower SNR: the analytical results are pessimistic. However, the insensitivity of the steady state normalized RSS to N is indeed as predicted by the analysis.

V. Conclusion

We have shown that, when we account for receiver noise, the one bit feedback algorithm does not scale indefinitely, in that the *fraction* of the ideal beamforming gain attained depends on per-node SNR, and does not improve with N. As we increase N, starting with a power pooling gain can certainly help enhance the reliability of communication right from the beginning of the process relative to a single node, but *normalized* progress towards the beamforming solution does not improve with N. This is because the progress in RSS over a step of the algorithm scales in the same manner as receiver noise, as shown by the covariance computations under our joint Gaussian approximations. We do note, however, the overall received SNR does improve monotonically with N, and in practical regimes of tens of cooperating nodes, significant performance gains (e.g., in terms of range extension while maintaining a reliable, spectrally efficient, link) can be obtained. It remains an open question as to whether there exist aggregate feedback algorithms, or hybrids of per-node and aggregate feedback, that can overcome the bottleneck identified in this paper.

Acknowledgments

This research was supported in part by the National Science Foundation under grant CCF-1302114, and by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office.

References

[1] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Distributed transmit beamforming using feedback control," *Information Theory, IEEE Transactions on*, vol. 56, no. 1, pp. 411–426, 2010.

[2] M. M. Rahman, H. E. Baidoo-Williams, R. Mudumbai, and S. Dasgupta, "Fully wireless implementation of distributed beamforming on a software-defined radio platform," in *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, ser. IPSN '12. New York, NY, USA: ACM, 2012, pp. 305–316. [Online]. Available: http://doi.acm.org/10.1145/2185677.2185745

[3] F. Quitin, M. M. U. Rahman, R. Mudumbai, and U. Madhow, "A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 3, pp. 1418–1428, 2013.

[4] F. Quitin, A. T. Irish, and U. Madhow, "A scalable architecture for distributed receive beamforming: analysis and experimental demonstration," *CoRR*, vol. abs/1501.05695, 2015. [Online]. Available: http://arxiv.org/abs/1501.05695

[5] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *Communications Magazine, IEEE*, vol. 49, no. 2, pp. 102–111, 2011.

[6] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "Airsync: Enabling distributed multiuser mimo with full spatial multiplexing," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 6, pp. 1681–1695, 2013.

[7] H. Rahul, S. Kumar, and D. Katabi, "MegaMIMO: Scaling Wireless Capacity with User Demands," in *ACM SIGCOMM 2012*, Helsinki, Finland, August 2012.

[8] J. Holfeld, E. Fischer, and G. Fettweis, "Field trial results for comp downlink transmissions in cellular systems," in *Smart Antennas (WSA), 2011 International ITG Workshop on*. IEEE, 2011, pp. 1–7.

[9] F. Quitin, A. Irish, and U. Madhow, "Distributed receive beamforming: A scalable architecture and its proof of concept," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, June 2013, pp. 1–5.

[10] R. Mudumbai, U. Madhow, R. Brown, and P. Bidigare, "Dsp-centric algorithms for distributed transmit beamforming," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, Nov 2011, pp. 93–98.

[11] P. Bidigare, M. Oyarzyn, D. Raeman, D. Chang, D. Cousins, R. O'Donnell, C. Obranovich, and D. Brown, "Implementation and demonstration of receiver-coordinated distributed transmit beamforming across an ad-hoc radio network," in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*. IEEE, 2012, pp. 222–226.