UCL DEPARTMENT OF
COMPUTER SCIENCE

Research Note
RN/13/19

FASTER: Fine and Accurate Synchronization for Large Distributed MIMO Wireless Networks

October 14, 2013

Konstantinos Nikitopoulos

Kyle Jamieson

Abstract

The recently-proposed AirSync and JMB systems allow spatially-separated transmitting radios to form a fully distributed multiple-input, multiple-output (MIMO) system. This makes dramatic wireless capacity gains possible even in networks where mobiles have just one to two antennas. However, very tight synchronization among transmitters' radio oscillators remains the limiting factor in these systems' performance. Even a slight loss in synchronization results in misalignment of the concurrently-transmitted signals and a consequent increase in the number of errored bits, harming network throughput. Furthermore, the demands of these systems for accurate synchronization increase monotonically with the number of participating access points (APs), and constellation density. So in order to maintain synchronization, these systems rely on periodic synchronization updates between all radios involved. This substantially limits capacity gains in the regime of tens of APs and high data rates.

We propose FASTER, a novel approach to synchronization that is orders of magnitude more accurate than that of the above systems. This frees a large distributed MIMO system from the need to sacrifice capacity by performing frequent phase updates. We have implemented FASTER in both simulation and on the Rice WARPv3 FPGA radio platform. Our experimental results show that FASTER achieves synchronization that is two orders of magnitude more precise than the best known practical approaches. FASTER is therefore the first practical synchronization algorithm to support distributed MIMO networks of tens of APs in size. It is also the first practical synchronization algorithm to support distributed MIMO in the presence of walking-speed client mobility.

This material is based on work supported by the European Research Council under Grant No. 279976.

FASTER: Fine and Accurate Synchronization for Large Distributed MIMO Wireless Networks

Konstantinos Nikitopoulos and Kyle Jamieson
University College London

ABSTRACT

The recently-proposed AirSync and JMB systems allow spatially-separated transmitting radios to form a fully distributed multiple-input, multiple-output (MIMO) system. This makes dramatic wireless capacity gains possible even in networks where mobiles have just one to two antennas. However, very tight synchronization among transmitters' radio oscillators remains the limiting factor in these systems' performance. Even a slight loss in synchronization results in misalignment of the concurrently-transmitted signals and a consequent increase in the number of errored bits, harming network throughput. Furthermore, the demands of these systems for accurate synchronization increase monotonically with the number of participating access points (APs), and constellation density. So in order to maintain synchronization, these systems rely on periodic synchronization updates between all radios involved. This substantially limits capacity gains in the regime of tens of APs and high data rates.

We propose FASTER, a novel approach to synchronization that is orders of magnitude more accurate than that of the above systems. This frees a large distributed MIMO system from the need to sacrifice capacity by performing frequent phase updates. We have implemented FASTER in both simulation and on the Rice WARPv3 FPGA radio platform. Our experimental results show that FASTER achieves synchronization that is two orders of magnitude more precise than the best known practical approaches. FASTER is therefore the first practical synchronization algorithm to support distributed MIMO networks of tens of APs in size. It is also the first practical synchronization algorithm to support distributed MIMO in the presence of walking-speed client mobility.

1. INTRODUCTION

One of the most fundamental challenges in modern wireless communication systems is to meet an ever-increasing demand for network throughput from ever-increasing numbers of users. One way of meeting this demand is the use of *spatial multiplexing*, whereby radios are equipped with multiple antennas (*i.e.*, MIMO). Then senders can

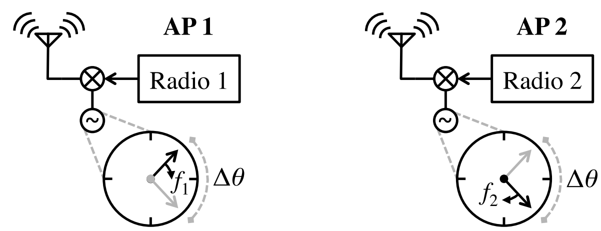

Figure 1: The oscillators of access points (APs) 1 and 2 have respective frequencies f_1 and f_2, resulting in a frequency offset $\Delta f = f_1 - f_2$. Between them they also have a separate but related time-varying phase offset of $\Delta\theta$, which is critical in determining overall system performance. Both APs send simultaneously using standard omnidirectional antennas.

stripe multiple streams of data concurrently over the same frequency band, increasing network capacity. But as many have observed [3, 5], the number of antennas at transmitter and receiver limits the gains that MIMO can achieve. Smaller mobile clients with their limited numbers of antennas exacerbate the issue.

A promising way of overcoming this limit, however, is to use antennas placed on different clients or access points (APs), distributing the MIMO system. And so there is an opportunity for such *distributed MIMO* systems to achieve dramatic, multiplicative increases in network capacity by allowing many clients to receive at the same time, over the same frequencies. In these systems, APs exchange data frames to be sent to each client over a wired Ethernet backhaul, then together send the data frames so that their combined transmissions corresponding to each client's data frame reinforce constructively at that client and destructively at other clients. This technique is called *transmit beamforming* when the system's goal is to deliver the data frame to a certain client [1, 14], and *interference alignment* when the system's goal is to reduce interference to others [5]. Recently, JMB [14] has realized distributed MIMO's potential to achieve throughput gains that scale linearly with the number of clients.

But how far can we push these capacity gains, in the face of increasing numbers of clients and increasing rates? The answer lies in a closer look at the factors involved in a distributed MIMO network's capacity apart from spatial multiplexing. These are the following:

1. Phase synchronization. Since the radios' oscillators are independent, each experiences a distinct oscillator frequency and phase offset (Δf and $\Delta\theta$, respectively) from every other transmitting AP or client, as Figure 1 shows for one pair of radios. Even small non-zero values of $\Delta\theta$ result in misalignment of the concurrent signals and a consequent increase in bit error rate (BER), harming throughput. Therefore the first step in the overall synchronization procedure is frequency synchronization, to drive Δf as low as possible.

Phase noise variation in $\Delta\theta$ cannot be compensated by any frequency synchronization algorithm, and therefore periodical phase updates are required. However in practice, while the phase rotation can significantly vary between adjacent samples (and therefore the instantaneous frequency can significantly change) the average phase rotation is usually constant for a relatively long time, For example, Murphy [11] shows that the average frequency can be unchanged for several tens of milliseconds. Consequently for such periods of time, phase noise can be well modelled as additive (thermal) noise.

2. Transmission rate. In the past, bit rate adaptation algorithms have adjusted the transmission rate on a link to match conditions, sending with a denser constellation when the wireless channel is strong, and a sparser constellation when the wireless channel is weak. Distributed MIMO systems' sensitivity to phase misalignments increases with the density of constellation symbols, and so in order to push capacity gains in these systems further, we need to drive $\Delta\theta$ close to zero.

Furthermore, in order to opportunistically capture instants when a weak channel briefly becomes strong, recent rateless transmission schemes send with a very dense constellation under all conditions [13]. Thus systems that attempt to combine distributed MIMO with rateless coding will place even greater demands on synchronization.

3. Resynchronization period. Even small errors in an estimated frequency offset will soon result in significant phase errors. Therefore, in order to maintain $\Delta\theta$ between all radios involved close to zero, a distributed MIMO system requires periodic phase resynchronization between radios, but this compromises channel efficiency. How frequently this needs to happen depends on both the uncertainty of the algorithm that estimates Δf, and second, the dynamics of Δf itself (which are a property of the oscillator). Therefore, further capacity gains in distributed MIMO systems can be met with the use of higher-precision oscillators, but this in turn demands higher-precision frequency estimation algorithms.

These three factors collude to prevent current distributed MIMO systems from fully achieving capacity gains commensurate with AP count past approximately 10 APs, as our evaluation shows (§3). In response, we propose a fundamentally different approach compared to previous systems. Instead of trying to estimate or track any phase differences due to inaccuracies of the frequency synchronization process as previous systems do, we seek to minimize Δf immediately and directly.

In this paper we present *FASTER*, a novel frequency synchronization algorithm that substantially outperforms prior work in terms of both its frequency estimation accuracy and computational complexity. FASTER achieves an accuracy close to the Cramer-Rao bound [6], more than two orders of magnitude better frequency estimation accuracy than standard approaches [16], while simultaneously requiring two orders of magnitude less computation than other algorithms of similar accuracy [10], whose computational overhead makes them difficult to realize in practical systems.

The key to FASTER's performance gains is that instead of attempting to measure phase rotation directly [10, 16], FASTER uses spectral (Fourier) analysis on the received preamble. However, standard spectral analysis is highly computationally demanding, so FASTER leverages a generalization of the classical Goertzel algorithm [17] to evaluate individual terms of a Fourier transform in a computationally-streamlined manner, thus iteratively refining its estimate of Δf.

The level of accuracy that FASTER achieves allows it to support denser modulations and larger numbers of simultaneously-sending APs without the need to rely on lengthening the synchronization preamble or averaging over many synchronization preambles. This in turn enables FASTER to better support mobile clients. In our performance evaluation (§3), we show that to achieve FASTER's rate and AP counts, the synchronization algorithms that systems such as JMB [14] use need to take measurements over such a long period of time that both the channel and Δf change, invalidating the estimate.

In the next section, we describe FASTER's design. A description of our implementation and a performance evaluation (§3) follows. We then discuss further related work in Section 4, and conclude (§5).

2. DESIGN

We now sketch the design of a generic distributed MIMO system, in order to put FASTER into context. In such systems, one AP acts as a *lead AP*, broadcasting periodic beacons, synchronizing the clients and other APs in time. Multiple APs exchange data and client channel estimates via a high-throughput backbone network (*e.g.*, Gigabit Ethernet, optical) and transmit on the downlink together,

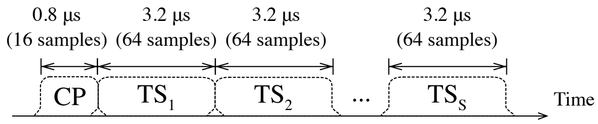

Figure 2: The training sequence added to the beginning of the lead AP's transmission consists of a short cyclic prefix followed by S identical training symbols (denoted "$\mathbf{TS_k}$" above).

forming a distributed MIMO downlink. Then, multiple clients transmit concurrently over the same frequency band to all APs, forming a distributed MIMO uplink.

Synchronization. The lead AP transmits a training sequence to all radios in the network consisting of S identical long OFDM training symbols, as shown in Figure 2. Then, upon receiving the training symbols, each other AP or client calculates its respective Δf relative to the lead AP. The effect of Δf is to rotate the complex received samples in time, and so correlation-based frequency estimation algorithms like Schmidl-Cox [16] calculate Δf by measuring the angle between pairs of received samples from $S = 2$ training symbols, which in the absence of frequency offset would be identical.

Even if the frequency synchronization is very accurate, after some (long) period of time the resulting phase offset will be large. This has been correctly noticed in JMB, and, therefore, we similarly assume that at the beginning of the frame phase update is required. However, as we will discuss later in detail, due to FASTER's, accuracy phase updates are less frequently required, or equivalently longer frames can be supported, resulting in a better channel utilization efficiency.

Another thing to notice is that, after the initial synchronization, we can use FASTER to further update the frequency estimate by using the preambles used for phase updates, as long as they these preambles are transmitted periodically. This practically increases the effective "training" period.

2.1 The FASTER algorithm

In contrast to the Schmidl-Cox and related algorithms which try to calculate Δf by measuring phase rotation, FASTER uses Fourier (spectral) analysis of the received training sequence. To accurately and reliably estimate Δf through Fourier analysis, we have two goals:

1. The input signal should have no other dominant frequency content except for Δf itself. In other words, the signal we analyze should be roughly constant in time except for the frequency offset component.
2. By the law of large numbers, larger averages have less variance (uncertainty), so we seek to collect energy from all the received samples, in order to minimize the uncertainty of our Δf estimate.

Let's denote the k^{th} received sample of the l^{th} training symbol as $r_l[k]$. Since the S OFDM symbols are 64 samples long, $k \in [1, 64]$ and $l \in [1, S]$.

Property 1 holds if we sample the training sequence shown in Figure 2 at a period of 64 samples, because the training symbols are identical and the entire sequence is short enough to go through a stationary channel at walking- or even driving-speed coherence times. With 64 S-point FFTs, we could therefore make 64 estimates of Δf and average them together:

$$
\begin{array}{lccccl}
\mathscr{F}(& r_1(1) & r_2(1) & \cdots & r_S(1) &) \rightarrow \widehat{\Delta f}_1 \\
\mathscr{F}(& r_1(2) & r_2(2) & \cdots & r_S(2) &) \rightarrow \widehat{\Delta f}_2 \\
 & & & \vdots & & \\
\mathscr{F}(& r_1(64) & r_2(64) & \cdots & r_S(64) &) \rightarrow \widehat{\Delta f}_{64}
\end{array}
$$

But if noise causes an error in any of these $\widehat{\Delta f}_k$, it will dominate the average and significantly diminish the precision of the estimate (Property 2).

Another approach might be to sum the received samples of each training symbol and perform spectral analysis on the result. However, since the phases of the received samples vary randomly due to the effect of the channel, such an addition can be destructive, as the curve labeled "Addition" in Figure 3 shows. How then can we combine all the received samples to make a more robust estimate in noise?

2.1.1 Coherent sample combination

The approach we take is to align all received samples to have the same phase as the first sample in each symbol, adding them *coherently*. In the absence of noise, the ratio $R_l(k) = \frac{r_l(k)}{r_l(1)}$ is constant over all l values, even for nonzero Δf. In the presence of noise, for each sample, we acquire an estimate of the above ratio ($\hat{R}(k)$) by adding the corresponding $R_l(k)$ across symbols. From this estimate we calculate the average phase difference between the kth sample and the first as $\hat{\phi}(k) = \angle \hat{R}(k)$. We are then in a position to coherently sum across the samples of each OFDM symbol to yield y_l ($l \in [1, S]$):

$$
\begin{array}{cccc}
r_1(1) & r_2(1) & \cdots & r_S(1) \\
r_1(2)e^{-j\hat{\phi}(2)} & r_2(2)e^{-j\hat{\phi}(2)} & \cdots & r_S(2)e^{-j\hat{\phi}(2)} \\
 & & \vdots & \\
r_1(64)e^{-j\hat{\phi}(64)} & r_2(64)e^{-j\hat{\phi}(64)} & \cdots & r_S(64)e^{-j\hat{\phi}(64)} \\
\hline
y_1 & y_2 & \cdots & y_S
\end{array}
$$

We then take an S-point FFT over the $[y_l]$ data. Choosing the maximum of the result yields a coarse initial frequency offset estimate $\widehat{\Delta f}_{initial}$. To see the gains of coherent combination against averaging we transmit a training sequence using the WARP, and after we correct its frequency offset, we observe its frequency spectrum. In Figure 3 we see that the maximum energy of the spectrum formed by

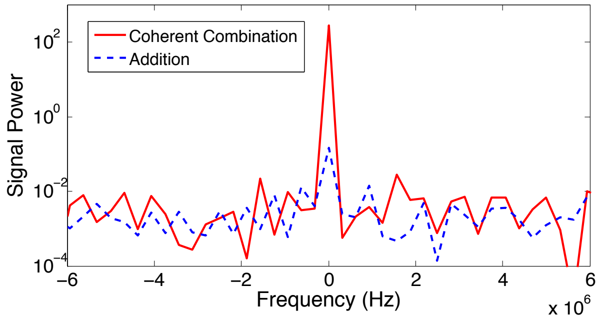

Figure 3: Spectrum with coherent combination and simple addition. The maximum energy of the coherent combination is orders of magnitude larger than the one of simple addition.

coherent combination is three orders of magnitude larger than that formed by simple averaging. Therefore, since spectral analysis methods choose the frequency with the maximum energy, estimators that use coherent combination are more robust to noise.

2.1.2 *Fine frequency offset estimation*

Our next challenge is performing a fine-grained, accurate spectral analysis on the $[y_l]$ data.

The spectral resolution of the FFT (and therefore the quality of the estimate) is a function of its size (N_{FFT}). In particular, if some frequency content lies between two FFT bins (*i.e.*, two integer multiples of $1/N_{FFT}$), then its power will be observed in the FFT bin whose frequency is closest to the signal. Therefore, the FFT introduces an uncertainty of half the FFT frequency bin size, or $1/2N_{FFT}$. To reduce this uncertainty we could increase the N_{FFT} and pad the data with zeroes in the time domain. However, as show later in Section 3, the required accuracy implies FFT sizes too large for practical hardware.

FASTER initially performs an FFT of size S (the number of training symbols). Then to refine $\widehat{\Delta f}_k$, it employs the Generalized Goertzel algorithm [17] which calculates the Discrete-Time Fourier Transform of a signal $x(\tau)$

$$X(\omega) = \sum_{\tau=-\infty}^{\infty} x(\tau) e^{-j\omega\tau}$$

at specific frequencies ω. For FASTER this becomes:

$$Y(\nu) = \sum_{l=1}^{S} y_l \cdot e^{-j\frac{2\pi}{S}\nu(l-1)}. \tag{1}$$

This can be calculated in two steps. First, we iteratively calculate

$$z_l = y_l + 2\cos\left(\frac{2\pi\nu}{S}\right) z_{l-1} - z_{l-2} \tag{2}$$

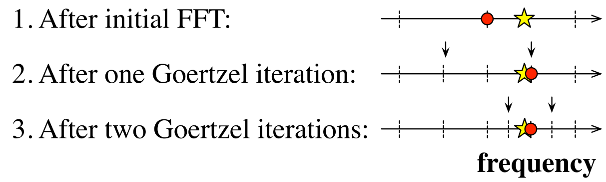

Figure 4: FASTER's "binary-search" frequency refinement algorithm: at each iteration a star indicates the true frequency offset and a circle indicates the frequency bin with greatest magnitude. Arrows show the new frequencies FASTER evaluates.

for $l = 1, ..., S+1$ and with $z_{-1} = z_{-2} = r_{S+1} = 0$. Then,

$$Y(\nu) = \left[z_{S+1} - e^{-j\frac{2\pi\nu}{S}} z_S \right] e^{-j2\pi\nu}. \tag{3}$$

The initial S-point FFT limits uncertainty to $\frac{1}{2S}$, so we use the Goertzel algorithm to evaluate the power spectrum of $[y_l]$ at frequencies $\widehat{\Delta f}_{initial} \pm \frac{1}{4S}$, thus obtaining a new frequency estimate with half the uncertainty. As shown in Figure 4, this process is repeated until we reduce the uncertainty to an acceptable level.

Finally, note that the receiver must estimate the wireless channel to the sender. We reuse the synchronization training sequence for this step, reducing overhead.

3. EVALUATION

In this section we measure the impact of synchronization on system throughput through simulations, then evaluate FASTER using simulations and our WARPv3 implementation. We compare with the following three algorithms:

1. JMB, which measures the phase of pairs of training symbols and then averages the resultant Δf estimates to calculate a final estimate. JMB spreads these pairs over different frames and therefore each training symbol is used only in one intermediate estimate.
2. A simple extension of the Schmidl-Cox algorithm [16] ("ESC") that estimates Δf repeatedly using overlapping pairs of adjacent training symbols, and averages the resultant estimates to calculate a final estimate.
3. A highly compute-intensive, non-linear approach proposed by Morelli and Mengali ("MM") [10] that can achieve near Cramer-Rao bound performance.

For all simulations we assume an eight-tap, independent and identically distributed (i.i.d.) Rayleigh channel. We simulate a distributed MIMO system using zero-forcing beamforming (similar to JMB). Packets are 1,500 bytes long, and the data are uncoded, 16-QAM modulated.[1]

3.1 The need for accurate synchronization

Impact on distributed MIMO system throughput. As discussed in Section 1, even small phase offsets result in

[1] Since a frequency synchronization error adds noise, similar results hold for all coding schemes and constellations.

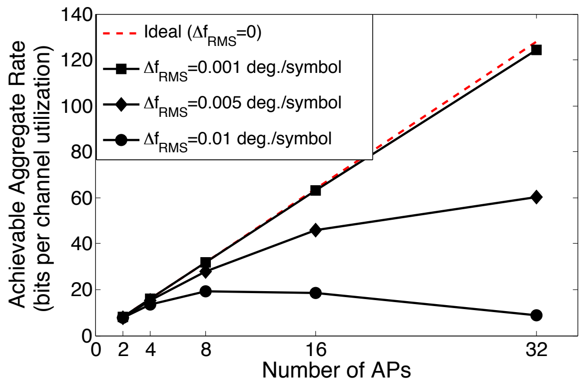

Figure 5: Effect of frequency estimation error on the achievable aggregate rate as function of the number of APs simultaneously transmitting 16-QAM and 1500-byte packets.

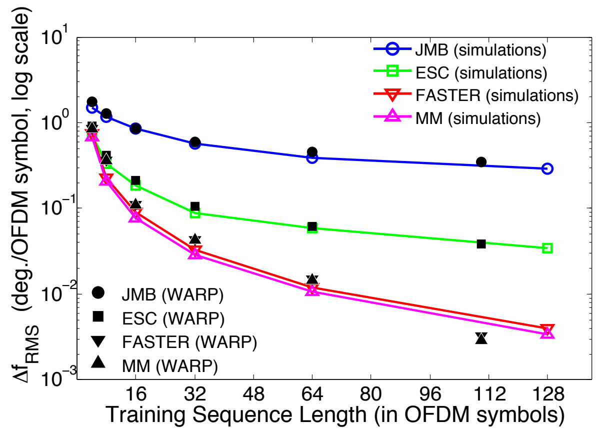

Figure 6: Estimation performance as a function of the training time. The SNR for simulations and WARP experiments is 13 and 13–16 dB respectively.

misalignment of concurrently transmitted signals and a consequent increase in BER, harming throughput. To quantify this effect, we simulate a distributed MIMO system's rate with the only impairment or noise being a randomly and normally distributed, zero-mean residual frequency offset of root-mean-square (RMS) error Δf_{RMS} (given in degrees per OFDM symbol) at the slave APs. Figure 5 shows that this aggregate rate markedly decreases with increasing frequency synchronization error. We also see that frequency offset estimation errors prevent us from using large AP counts, and that if we want to increase the number of APs from four to 32, we need an order of magnitude more accurate frequency estimation.

Frequency oscillator drift. As discussed in Section 2, to achieve such a level of accuracy we can use long training synchronization sequences. However, reaching high levels of accuracy requires that both the transmission channel and the oscillators' frequency remain static over the training sequence period. This is tens of milliseconds for walking-speed, 2.4 GHz systems. Even in the absence of mobility, the dynamics of Δf limit the size of the training sequence that we can use efficiently. Specifically, when the training sequence is so long that Δf changes within its length by an amount greater than the uncertainty of the estimator itself, the frequency estimate error will be dominated by the corresponding dynamics.

Murphy [11] has quantified these frequency dynamics for the WARP platform.[2] He shows that within a timescale of 100 ms the frequency changes randomly with a constant mean and an RMS variance of less than 4 Hz (which for a 20 MHz, 64-subcarrier system corresponds to about 0.006°/symbol). Within a period of a 500 ms we can see an additional change in the mean value of about 5 Hz, or an RMS error of roughly 0.013°/symbol, while within two seconds we can see changes of more than 15 Hz or 0.022°/symbol. This highlights that oscillator frequency dynamics can be the critical limiting factor in distributed MIMO performance, especially when the system uses large training sequences. In the next section we look at the training sequence length that each system requires.

3.2 FASTER evaluation

Estimation accuracy. We compare FASTER's estimation accuracy against JMB, ESC, and MM algorithms; both by simulation and by implementation on the WARP platform at 20 MHz bandwidth. Using WARPLab we transmit 110 consecutive training symbols. Then, at the receiver, we perform frequency estimation with each of the examined methods, using S long training symbols from the data.[3] Then, we compare these frequency estimates with the corresponding estimates acquired over the whole, oversampled (at 40 MHz) received sequence which is a much stronger estimate due to the longer training sequence, and therefore we take it as the "ground-truth." We perform the experiments for an SNR range between 13–16 dB. For the simulation we assume an SNR of 13 dB. In Figure 6 we first validate that FASTER can reach the near-optimal performance of MM. In addition, FASTER significantly outperforms ESC and JMB. In particular, for $S = 128$, FASTER is an order of magnitude more accurate than ESC and two orders of magnitude than JMB, with very close agreement between simulation and experiments.

Complexity. In Figure 7 we show the complexity of the different approaches in terms of complex multiplications.

[2]The WARP platform uses a Crystek CVT32 crystal clock oscillator and MAX2829 transceiver [11], which are representative of the most recent commercial 802.11n/ac chipsets' higher frequency stability.

[3]For a conservative comparison with JMB, we do not incorporate OFDM cyclic prefix overhead into S, but JMB's non-contiguous training sequences require more cyclic prefixes.

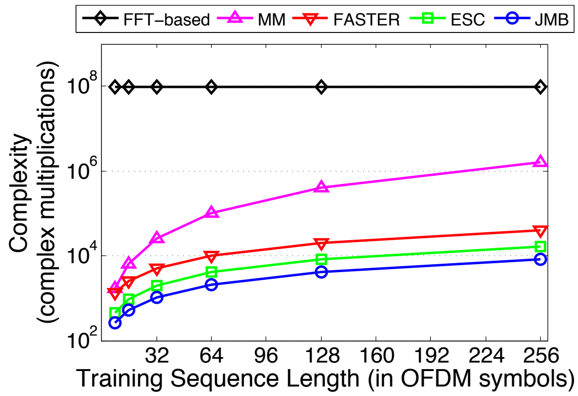

Figure 7: Complexity comparison of the different frequency estimation methods.

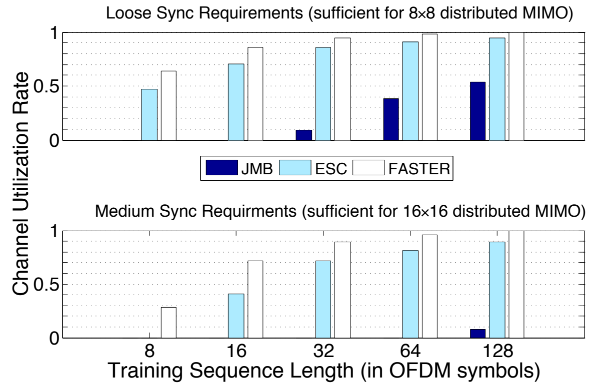

Figure 8: Channel utilization rate for different estimators, tight and medium synchronization requirements, 13 dB SNR, 16-QAM and 1500-byte packets.

We set the required uncertainty of FASTER less than 10^{-4} degrees per OFDM symbol. We also show the complexity for the same target accuracy if instead of FASTER we perform spectral analysis by using a large FFT (and zero padding). We see that FASTER is more than an order of magnitude less complex than the MM, and approximately four orders of magnitude less complex than the FFT, without compromising the estimation performance. The JMB and ESC algorithms are less complex but, as we showed in Figure 6, their accuracy is orders of magnitude poorer than the one of FASTER and MM.

Channel Utilization and Resynchronization. As we discussed in Section 1 even small errors in an estimated frequency offset will soon result in significant phase errors. Therefore, in order to maintain a small phase error, we can periodically phase resynchronize based on preambles sent by the lead-AP. Unfortunately, such transmissions reduce the available time for data transmission, and therefore reduce the achievable throughput. We now evaluate how efficiently the channel can be utilized as a function of the frequency synchronization method and the required phase synchronization accuracy. Regarding the phase accuracy, we examine two scenarios. The "loose sync requirements" scenario assumes that the maximum phase rotation due to residual frequency error should never be larger than the RMS rotation in the end of 1500-byte frame with $\Delta f_{RMS} = 0.01°$/symbol (without any resynchronization). According to Figure 5 this level of accuracy enables a nearly linear relationship between throughput and AP count for up to eight APs. The second, "medium sync requirements" scenario assumes a maximum frequency error of $\Delta f_{RMS} = 0.005°$/symbol. According to Figure 5 this level of accuracy enables a nearly linear relationship between throughput and AP count for up to 16×16 distributed MIMO. For our calculations we assume that perfect phase synchronization can be achieved transmitting preambles of a single OFDM symbol length. In reality, longer preambles may be needed to acquire accurate phase estimates in noisy environments, which will result in further throughput degradation than the ones we show due to phase updates. In this case, the accuracy of the synchronization algorithm will be even more significant. For our evaluation we use the results of Figure 6 for an SNR of 13 dB. Figure 8 shows that FASTER significantly outperforms all other approaches with gains which increase when the synchronization requirements become tighter. We note that depending on the accuracy of the estimate, the frequency error could be such that resynchronization is required every OFDM symbol, resulting in a complete degradation of channel utilization and throughput.

Explaining JMB's linear throughput gains. The natural question that arises after the previous discussion is why JMB achieves its published throughput gains, which scale linearly with the number of clients. To explain this we run some additional simulations evaluating Δf_{RMS} (Figure 6) for 13 dB SNR and even larger training sequences. In order for JMB to achieve a Δf_{RMS} accuracy of 0.01°/symbol (sufficient for a close-to-linear throughput increase for a 8×8 system), it requires a training sequence of 6.5×10^4 symbols (about 210 ms for a system at 20 MHz). In indoor static environments where coherence time is hundreds of milliseconds ([14], §5), JMB therefore excels.

Then why do we need FASTER? In order to achieve a Δf_{RMS} accuracy of 0.004 (sufficient for a 16×16 system) at 13 dB, FASTER requires 128 OFDM symbols (410 μs at 20 MHz), and JMB requires 5×10^5 OFDM symbols (1.7 seconds at 20 MHz), which exceeds an indoor stationary channel coherence time. In addition, as Murphy [11] shows, frequency oscillator changes would reach more than 0.017°/symbol, preventing JMB from reaching the accuracy of FASTER in this regime. This therefore precludes JMB from scaling in large distributed MIMO systems with 16 APs or more, or being used in mobile environments where the coherence time can be down to a few milliseconds, as in the case of walking-speed client mobility in the 5 GHz band.

4. RELATED WORK

Two recent distributed MIMO systems, JMB [14] and AirSync [1], have demonstrated in practice that such systems can achieve dramatic increases in network throughput. We have discussed JMB at length in the preceding, so focus on AirSync in this section.

AirSync achieves tight synchronization by exploiting full-duplex wireless communication. A lead AP broadcasts pilot tones, while the other APs receive these tones while transmitting and use Kalman filtering to phase-synchronize their own transmissions. The lead AP, however, transmits these tones outside the data band, consuming additional bandwidth. In addition, the Kalman filter's convergence time limits synchronization accuracy in the presence of frequency offset dynamics.

Since frequency synchronizaion is one of the most important tasks performed in wireless receivers, the corresponding literature is very rich. However, it typically focuses on how to extend the frequency estimation range and estimation performance when using one or maximum two OFDM symbols. FASTER, goes one step further. It applies to long training sequences which are required when the estimation accuracy provided by short training sequences is insufficient, as in the case of distributed MIMO systems. Typical frequency synchronization techniques ([9, 10, 16]) employ short preambles (one or two OFDM symbols) that consist of identical parts. Estimation of the frequency offset is then performed by calculating phase rotations between these identical parts. Except for the MM algorithm which we have extensively discussed, such methods are not applicable large training sequences or to very large frequency offsets. This is because they require phase rotation, due to frequency offset, between identical parts should not exceed 2π. To avoid this problem and to reduce computational complexity, Cvetkovic *et al.* have recently proposed in [4] to perform phase unwrapping and to perform estimation using the phases of the received samples instead of their complex values. While Cvetkovic proposed the algorithm for short training symbols (i.e., one OFDM symbol), it is also applicable to long training sequences. However, the main drawback of the approach is that its complexity is a function of the frequency offset, and can thus explode for small frequency offsets. In particular, its maximum complexity when applied to FASTER's sequence is $\mathcal{O}(64 * S^2)$ which, as we see in Figure 7, is similar to the complexity of the MM algorithm. Instead of using known preambles, Van de Beed *et al.* [18] suggest to exploit the redundancy of the cyclic prefix over data transmission while Bolcskei [2] suggests to explore the statistics of the received data OFDM signal. However, to apply such approaches in a fully distributed MIMO system, the slave APs should be able to receive the signal from the lead AP while transmitting to the clients. In other words, and in contrast to FASTER, such techniques require APs with full-duplex capabilities.

Similarly to FASTER, Lei and Ng [7] perform spectral analysis to estimate the frequency offset. In particular, they show we can find can find the maximum-likelihood frequency estimate via spectral analysis. However, in order to cope with the complexity of calculating the periodogram, and avoid zero-padding, they propose a suboptimal algorithm which requires a specific pilot structure (*i.e.*, an OFDM symbol with distinctively spaced pilots), which prohibits its use for concurrent synchronisation and highly accurate channel estimation (*i.e.*, it would require frequency domain interpolation), as is feasible with FASTER. FASTER, on the other hand, is applicable to Lei and Ng pilot structure. In addition, the algorithm of Lei and Ng uses only a single OFDM symbol, and its extendability to long training sequences is not examined. Such an extension is feasible by means of FASTER's techniques. In [8] Li *et al.* also perform spectral analysis but they use aggressive zero-padding to compute the periodogram up to the required accuracy which, as we discussed, is not efficient for highly accurate estimators.

Finally, methods to compensate for the effects of the residual frequency offset at the receiver side have been proposed in [12, 15]. However, such approaches cannot compensate for the effects of erroneous precoding, but only for distinct syncronization errors between APs and clients.

5. CONCLUSION

We have described FASTER, a synchronization algorithm for distributed MIMO wireless networks. FASTER's direct and highly-accurate frequency offset estimate enables distributed MIMO at scales greater than were possible before, with respect to transmission rate, number of APs, and number of users.

6. REFERENCES

[1] H. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire. Achieving high data rates in a distributed MIMO system. In *Proc. of MobiCom*, 2012.

[2] H. Bolcskei. Blind estimation of symbol timing and carrier frequency offset in pulse shaping ofdm systems. *IEEE Trans. Commun.*, 49(6):988–999, 2001.

[3] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran. Multiuser MIMO achievable rates with downlink training and channel state feedback. *IEEE Trans. on Information Theory*, 56(6):2845–2866, 2010.

[4] Z. Cvetkovic, V. Tarokh, and S. Yoon. On frequency offset estimation for ofdm. *IEEE Trans. on Wireless Commun.*, 12(3):1062–1072, 2013.

[5] S. Gollakota, S. Perli, and D. Katabi. Interference alignment and cancellation. In *Proc. of ACM SIGCOMM*, Aug. 2009.

[6] S. M. Kay. *Fundamentals of statistical signal processing, Volume I: Estimation Theory (v. 1)*. Prentice Hall, 1993.

[7] J. Lei and T.-S. Ng. A consistent ofdm carrier frequency offset estimator based on distinctively spaced pilot tones. *IEEE Trans. Wireless Commun.*, 3(2):588–599, 2004.

[8] J. Li, G. Liu, and G. Giannakis. Carrier frequency offset estimation for ofdm-based wlans. *IEEE Signal Processing Letters*, 8(3):80–82, 2001.

[9] P. H. Moose. A technique for orthogonal frequency division multiplexing frequency offset correction. *IEEE Trans. Commun.*, 42(10):2908–2914, 1994.

[10] M. Morelli and U. Mengali. An improved frequency offset estimator for ofdm applications. In *Communication Theory Mini-Conference*, pages 106–109, Jun. 1999.

[11] P. Murphy. *Design, Implementation and Characterization of a Cooperative Communications System*. PhD thesis, Rice University, 2010.

[12] K. Nikitopoulos and A. Polydoros. Phase-impairment effects and compensation algorithms for ofdm systems. *IEEE Trans. Commun.*, 53(4):698 – 707, April 2005.

[13] J. Perry, P. Iannucci, K. Fleming, H. Balakrishnan, and D. Shah. Spinal codes. In *Proc. of ACM SIGCOMM*, 2012.

[14] H. Rahul, H. Hassanieh, and D. Katabi. JMB: Scaling wireless capacity with user demands. In *Proc. of ACM SIGCOMM*, 2012.

[15] B. Sahu, S. Chakrabarti, and S. Maskara. An improved residual frequency offset estimation scheme for {OFDM} based {WLAN} systems. *Elsevier, Digital Signal Processing*, 20(2):454 – 461, 2010.

[16] T. Schmidl and D. Cox. Robust frequency and timing synchroniation for OFDM. *IEEE Trans. on Communications*, 45(12):1613–1621, Dec. 1997.

[17] P. Sysel and P. Rajmic. Goertzel algorithm generalized to non-integer multiples of fundamental frequency. *EURASIP Journal on Advances in Signal Processing*, 2012:1–8, 2012.

[18] J.-J. van de Beek, M. Sandell, and P. Borjesson. Ml estimation of time and frequency offset in ofdm systems. *IEEE Tran. Signal Processing*, 45(7):1800–1805, 1997.