

Poster: Clock Synchronization for Distributed Wireless Protocols at the Physical Layer

Omid Abari, Hariharan Rahul, and Dina Katabi
Massachusetts Institute of Technology
Cambridge, MA, USA
abari@csail.mit.edu, rahul@csail.mit.edu, dina@csail.mit.edu

SUMMARY- Implementing distributed wireless protocols at the physical layer today is challenging because different nodes have different clocks, each of which has slightly different frequencies. This causes the nodes to have frequency offset relative to each other. As a result, transmitted signals from these nodes do not combine in a predictable manner over time. Past work tackles this challenge and builds distributed PHY layer systems by attempting to address the effects of the frequency offset and compensating for it in the transmitted signals. In this poster, we address this challenge by addressing the root cause - the different clocks with different frequencies on the different nodes. We present AirClock, a new wireless coordination primitive that enables multiple nodes to act as if they are driven by a single clock that they receive wirelessly over the air. AirClock presents a synchronized abstraction to the physical layer, and hence enables direct implementation of diverse kinds of distributed PHY protocols. We illustrate AirClock's versatility by using it to build two systems: (1) distributed MIMO, and (2) distributed rate adaptation for wireless sensors, and show that they can provide significant performance benefits over today's systems.

CATEGORIES AND SUBJECT DESCRIPTORS

C.2.1 [Computer-Communication Networks]: Network Architecture and Design

KEYWORDS

Clock Synchronization; Distributed MIMO; Wireless; Wireless Sensors

1. INTRODUCTION

Distributed processing is common at higher layers of the network stack. Yet, at the physical layer, each node typically performs its signal processing independently of other nodes in the network. This architecture is despite the benefits from distributed computation. In fact, the ability to coordinate nodes at the physical layer brings dramatic benefits. Consider, for *e.g.*, distributed MIMO, where multiple wireless nodes coordinate their transmissions to emulate one MIMO transmitter with the sum of antennas on all nodes. Such a design scales wireless throughput with the number of transmit-

ters [10, 1]. In fact, information theory presents many theoretical distributed physical layer designs that could bring significant gains if transformed into practical systems, such as distributed lattice coding [8], noisy network coding [6], transmitter cooperation for cognitive networks [7], etc.

The difficulty in building practical distributed PHY systems is fundamental. Theoretical schemes typically require that the different wireless nodes operate in lockstep with each other, so that the combined output of the wireless system generated from the computations and actions of individual wireless nodes follows a predictable pattern. However, independent wireless nodes have different clocks with different oscillator frequencies. As a result, different nodes will always have an offset in their carrier frequencies (CFO); the CFO causes two signals to rotate with respect to each other. The output signals formed by the distributed wireless system, therefore, no longer follows the desired pattern; for example, signals that are coordinated to be in-phase and combine constructively could become completely out-of-phase over time, and cancel each other. While having a consistent notion of time is important for distributed protocols both at higher and lower layers, the problem becomes significantly harder at the physical layer, where tiny timing errors of few nanoseconds or even less can cause large errors in the distributed protocol [10].

So, why do different nodes have different clock and oscillator frequencies? Today, each node has a local crystal that generates a low frequency sine wave, which is used as a reference clock. A special circuit called a Phase Locked Loop (PLL) uses the reference to generate the desired carrier frequency. For example, if the node is a Wi-Fi radio, the crystal might generate a 10 MHz sine wave, which the PLL then upconverts to a center frequency in the 2.4 GHz or 5 GHz bands. The key problem is that reference clocks on different nodes have slightly different frequencies due to different crystals and temperature changes. Since the PLLs on different nodes lock to reference clocks with different frequencies, their output signals have different frequencies, and this leads to frequency offsets between nodes.

2. OUR SOLUTION: AIRCLOCK

This abstract explores a solution that addresses the *root cause* of the problem, *i.e.*, the different reference clocks. It does so by replacing these different clocks with a single reference clock that is shared across all the nodes. If wireless nodes were driven by the same reference clock, they would have the same oscillator frequencies, and it would eliminate much of the complexity in building distributed PHY protocols.

One could of course share the reference among multiple wireless nodes by using wires from a single external clock. However, such a system defeats the notion of a wireless network since all nodes are now wired to a central device!

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).

MobiCom'14, September 7-11, 2014, Maui, Hawaii, USA.

ACM 978-1-4503-2783-1/14/09.

<http://dx.doi.org/10.1145/2639108.2642894>.

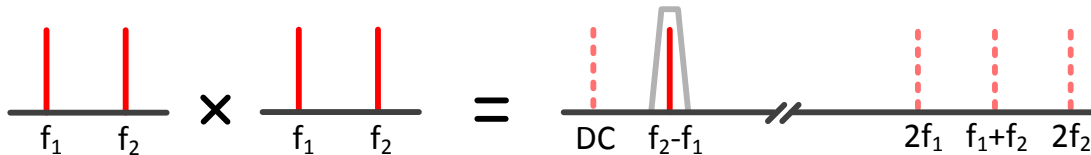


Figure 1—AirClock’s Design. Nodes multiply received signal by itself and extract the clock by applying a bandpass filter centered at f_{ref} .

We therefore focus on a design that feeds the shared reference clock to multiple nodes across the wireless medium. At first blush, it might seem that we can simply wirelessly transmit the low frequency wave previously generated by the local crystal on each node, and use it as a shared reference. However, such a system is not feasible for two reasons. First, transmitting at such low frequencies (typical crystal generated frequencies are in the range of 10-40 MHz) does not comply with FCC regulations [4]. Second, efficient antennas to capture low frequency signals can be large [2] and, therefore, impractical for many wireless nodes.

We introduce AirClock, a practical system that enables distributed wireless nodes to share a reference clock using the wireless medium. AirClock leverages the recently opened white spaces to transmit two single tones separated by the desired clock frequency (e.g., for a clock of 10 MHz, AirClock can send tones at 175 MHz and 185 MHz). To obtain the shared clock, each wireless node multiplies the received signal by itself and applies a band pass filter to extract the desired clock frequency. To see why this system works, recall that the multiplication of two sine waves at different frequencies produces sine waves whose frequencies are the sum and difference of the original frequencies. Specifically, if the two transmitted tones are at frequencies f_1 and f_2 , and the desired clock has a frequency of $f_{ref} = f_1 - f_2$, we have:

$$\begin{aligned}
 &(\cos(2\pi f_1 t) + \cos(2\pi f_2 t)) \cdot (\cos(2\pi f_1 t) + \cos(2\pi f_2 t)) = \\
 &1 + \mathbf{\cos(2\pi f_{ref} t)} + \cos(2\pi(f_1 + f_2)t) \\
 &+ \frac{1}{2} \cos(2\pi(f_1 + f_1)t) + \frac{1}{2} \cos(2\pi(f_2 + f_2)t)
 \end{aligned}$$

Then, by applying a band pass filter centered at f_{ref} , a node can extract the desired clock signal (the term highlighted in bold) and eliminate the undesired frequencies in the signal.¹ Fig. 1 illustrates this concept.

Since all nodes extract the reference clock only from the received signal without using any other external signals, they all extract the same reference frequency, thereby eliminating frequency offsets between nodes.

AirClock has several key features. (a) It is **versatile**. Since AirClock transparently provides a synchronized clock abstraction across multiple independent nodes, it enables straightforward implementation of very different kinds of distributed PHY schemes without application or protocol specific mechanisms. (b) Its design is **simple**. Since AirClock only needs to generate a sine wave as a reference clock, it only needs an amplifier, mixer, and filter, and avoids most of the complexity of a typical receiver, such as ADCs, baseband processing such as demodulation, equalization, decoding *etc.* (c) It easily **integrates with standard transceiver architectures**. Specifically, it can be deployed as a standalone module

¹Note that one cannot share the reference clock by simply upconverting the 10 MHz to a higher frequency at the transmitter and downconverting at the receiver to retrieve the reference clock. This is because downconversion would require the receiver to have a local oscillator to generate the carrier frequency, which would lead to a frequency offset in the retrieved clock signal.

whose output can be plugged in as an input to the PLL, replacing the output of the local crystal.

3. APPLICATIONS OF AIRCLOCK

AirClock is versatile and can be used as a synchronization primitive underlying various distributed PHY systems. Below, we describe two examples:

(a) Distributed MIMO without phase coordination: Distributed MIMO is a powerful concept that allows multiple wireless transmitters to behave like one huge MIMO transmitter with the number of antennas equal to the sum of antennas on all the cooperating nodes. These distributed transmitters can use AirClock as a primitive to synchronize their oscillators, and ensure coherent phase transmission. Such a system eliminates the need for a coherent phase transmission protocol in baseband [10, 1].

(b) Distributed Rate Adaptation for Sensors: Traditional sensor systems such as Zigbee [3] are typically capable of using only a single low rate, and hence cannot exploit good channel conditions to improve throughput. Distributed rate adaptation schemes that allow multiple nodes to transmit simultaneously and thereby exploit good channel conditions, have been designed and implemented for backscatter networks such as RFIDs [11]. However, such schemes do not work for wireless sensors since, unlike RFIDs, sensors have independent local oscillators, and hence suffer from frequency offsets. By using AirClock, sensor networks can eliminate frequency offsets between nodes, enabling them to leverage backscatter distributed rate adaptation algorithms. We design and implement the first practical distributed rate adaptation system for sensor networks.

4. RESULTS SUMMARY

We built a prototype of AirClock using off the shelf components and integrated it with USRPs to demonstrate distributed PHY applications. We evaluated AirClock and its applications in an indoor testbed with line-of-sight and non line-of-sight scenarios. Our results show the following:

- AirClock can provide tight synchronization across multiple nodes. In particular, in our testbed, the median and 95th percentile CFO between two AirClock nodes operating at 2.45 GHz are 0.38Hz (0.16 parts per billion) and 1.24Hz (0.5 parts per billion) respectively. AirClock can thus achieve orders of magnitude tighter synchronization than the traditional situation of free running oscillators on independent nodes.
- To investigate the impact of AirClock’s synchronization on the received signal, we transmit OFDM packets using 64-QAM modulation. We then estimate the received signal after channel equalization on the receiver. We repeat the experiment with three different synchronization modes between transmitter and receiver nodes: (a) independent clocks (*i.e.*, local crystals), (b) single external clock shared over a wire between the nodes, and (c) AirClock synchronization at both nodes. Fig. 2 plots the received constellation. We see that the transmit and receive oscillators have significant CFO with each other when using their local

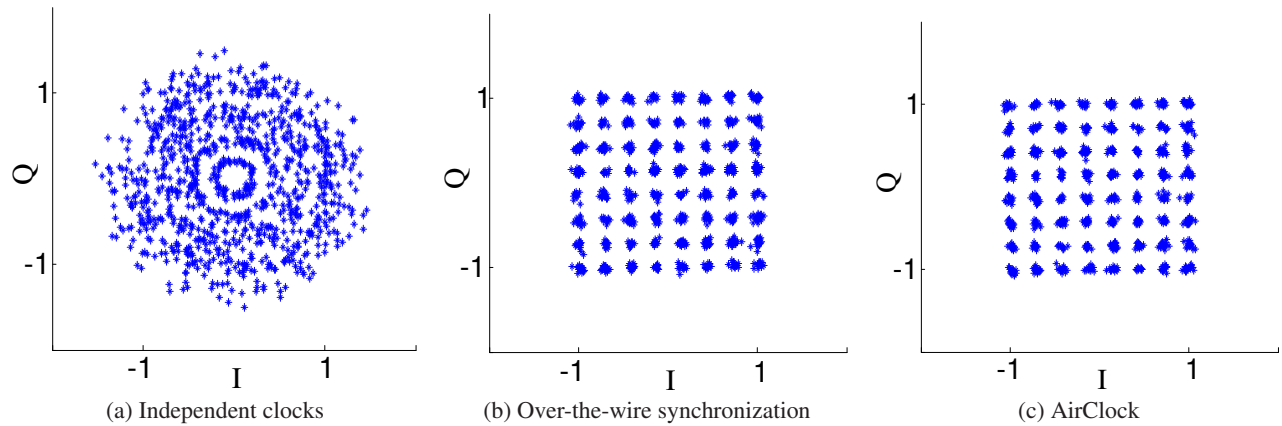


Figure 2—Received constellation points for 64-QAM in different synchronization scenarios. With different crystals, the transmitter and receiver are not synchronized, leading to rotation of the constellation points. In contrast, while using over-the-wire synchronization, the received constellation points after equalization do not rotate relative to each other, showing that the transmitter and receiver oscillators are synchronized. AirClock provides performance similar to over-the-wire synchronization by using a wireless reference signal.

crystals. As a result, the observed constellation rotates over time. In contrast, the observed constellation points with AirClock are tightly clustered around the expected points, similar to the case when both nodes are connected to a single external clock over the wire. This provides visual evidence that AirClock provides tight synchronization comparable to a shared wired reference.

- AirClock delivers a distributed MIMO system whose throughput scales linearly with the number of transmitters. Specifically, with 5 transmitters, AirClock’s distributed MIMO delivers a median throughput gain of $4.4\times$ over traditional 802.11 style unicast, across the operational range of Wi-Fi SNRs (5-20 dB).
- An AirClock based Zigbee system achieves throughput gains over traditional Zigbee by using distributed rate adaptation. Specifically, with 6 Zigbee nodes transmitting jointly to a central node, AirClock provides throughput gains of $1.64 - 3\times$ across a range of SNRs (5-20 dB).

5. RELATED WORK

(a) Shared clocks: One traditional solution to synchronize clocks in a wireless network is to use global positioning system (GPS)-disciplined oscillator to generate the reference signal [5]. Due to their complexity, these clock generators cost thousands of dollars which makes them infeasible for use in each wireless node. In addition, GPS only works outdoors. In contrast, AirClock does not require expensive hardware and can work in indoor applications.

Also, terrestrial time signals such as WWV [9], transmitted by NIST, can synchronize receivers on the order of seconds. Similarly, traditional clock synchronization schemes like NTP can synchronize nodes to within 10s of msec. However, unlike AirClock, these schemes are too coarse grained for PHY systems, which require synchronization on the order of nanoseconds.

(b) Phase tracking and compensation: The typical approach for dealing with frequency offset in wireless nodes is to estimate the resulting phase rotation and compensate for it. For distributed MIMO, schemes such as MegaMIMO and AirSync [1, 10] incorporate sophisticated schemes to track transmitter phase drift and correct signals prior to transmission. These schemes are necessary to ensure that the signals from different transmitters combine coherently at each receiver. In contrast, with AirClock, wireless nodes synchronizes their clocks over the air eliminating the need for phase tracking and compensation either at the receiver or other transmitters and can achieve both diversity and multiplexing gain. Further, AirClock’s system is independent of the protocol, and modulation

schemes, and applies to various wireless technology (WiFi, Bluetooth, Zigbee, etc.).

6. CONCLUSION

This extended abstract presents AirClock, a system that can synchronize multiple nodes using a reference signal that they receive over the wireless medium, and describes distributed PHY applications that can be built using AirClock’s synchronization primitive. We believe that AirClock’s synchronization primitive can serve as a building block that brings a large body of distributed information theoretic schemes closer to practice.

REFERENCES

- [1] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire. Airsync: Enabling distributed multiuser mimo with full spatial multiplexing. 2012.
- [2] C. A. Balanis. *Antenna theory: analysis and design*. John Wiley & Sons, 2012.
- [3] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. *Computer communications*, 30(7):1655–1695, 2007.
- [4] FCC, “FCC online table of frequency allocation, April 16, 2013”.
- [5] Jackson Labs, Fury GPSDO. <http://jackson-labs.com/>.
- [6] S. H. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung. Noisy network coding. *IEEE Transactions on Information Theory*, 57(5):3132–3152, 2011.
- [7] I. Maric, N. Liu, and A. Goldsmith. Encoding against an interferer’s codebook. In *46th Annual Allerton Conference on Communication, Control, and Computing, 2008*, pages 523–530. IEEE, 2008.
- [8] B. Nazer and M. Gastpar. The case for structured random codes in network capacity theorems. *Eur. Trans. on Telecommunications*, 19(4):455–474, 2008.
- [9] NIST. Radio Station WWV. <http://www.nist.gov/pml/div688/grp40/wwv.cfm/>.
- [10] H. Rahul, S. S. Kumar, and D. Katabi. MegaMIMO: Scaling wireless capacity with user demand. In *SIGCOMM*, 2012.
- [11] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and reliable low-power backscatter networks. *ACM SIGCOMM Computer Communication Review*, 42(4):61–72, 2012.