Design and Implementation of Scalable Massive-MIMO Networks

Clayton Shepard[†‡], Josh Blum[‡], Ryan E. Guerra[‡], Rahman Doost-Mohammady[†], and Lin Zhong[*†]

[†]Rice University [‡]Skylark Wireless [*]Yale University

ABSTRACT

Massive-MIMO technology promises many-fold improvement in the capacity of wireless mobile networks by scaling up the number of base station antennas. Hence, it has become a key technology for upcoming fifth generation (5G) mobile networks. Previous work has mainly focused on building proof-of-concept systems to show the feasibility of massive-MIMO and its benefits. However, much work remains to demonstrate the performance of massive-MIMO in at-scale dense deployments. In this paper, we present our experience in building ArgosNet, a scalable multi-cell campus-wide massive-MIMO testbed. The goal of ArgosNet is to provide scalability, programmability, and ease of deployment for this technology in realistic large-scale environments. We present the design choices we made in architecting ArgosNet and provide details in how they helped us achieve these goals. By characterizing the performance of major components, we discuss how ArgosNet unlocks our ability to do research that is not possible with existing systems.

CCS CONCEPTS

• **Hardware → Beamforming**.

KEYWORDS

Many-Antenna, Massive-MIMO, Large-scale Antenna Systems (LSAS), MU-MIMO, Distributed-MIMO, Network-MIMO, Beamforming

ACM Reference Format:
Clayton Shepard[†‡], Josh Blum[‡], Ryan E. Guerra[‡], Rahman Doost-Mohammady[†], and Lin Zhong[*†]. 2020. Design and Implementation of Scalable Massive-MIMO Networks. In *The 1st International Workshop on Open Software Defined Wireless Networks (OpenWireless'20), June 19, 2020, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3396865.3398691

1 INTRODUCTION

Massive-MIMO technology allows high degrees of spatial spectrum reuse through multi-user beamforming. Theoretically, as the number of antennas at the base station increases, the higher number of concurrent user streams are enabled by leveraging spatial multiplexing gains.

In this paper, we share our experience building ArgosNet, a network testbed consisting of multiple cooperating massive-MIMO base stations and their clients. ArgosNet draws lessons from our experience of building multiple generations of Argos massive-MIMO base stations [27–29].

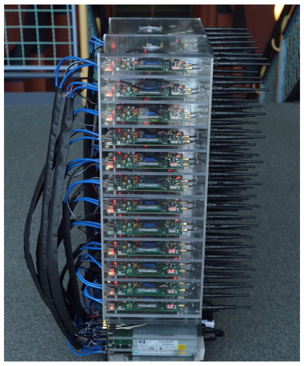
ArgosV1 was the first publicly reported massive-MIMO base station prototype (64 antennas) [28]. It empirically solves the problem of scaling up the *base station*. When building ArgosV1, our primary goal was to demonstrate the feasibility of massive-MIMO base stations using commercial software-defined radio hardware, i.e., WARPv1, and to experimentally evaluate the theoretical benefits of massive-MIMO, as predicted by [13, 14]. ArgosV2 empirically solves the problem of scaling up MU-MIMO *systems*, not just the base station, with a novel control channel design that bridges the gain gap [27]. ArgosV2 employed newly available WARPv3 [21]. It was the first publicly reported real-time massive-MIMO base station prototype and set a new record with 108 radios.

Funded by an NSF CRI grant, ArgosNet was to build a *network* testbed with multiple massive-MIMO base stations so that networking challenges such as inter-cell interference and cooperation can be experimentally studied [25]. This goal led us to pursue the following concrete objectives:

- Support for extreme scalability to enable coherent MU-MIMO experiments with 500 or even 1000+ active antennas, creating a truly *massive* system.
- Frequency agility and *dynamic* spectrum access support, from UHF to the emerging 3.5 GHz CBRS band.
- Support for more widely-used *open-source* SDR flows such as GNURadio [4] or OpenAirInterface [17] that rely on continuous streaming of samples from hardware.
- Support for *at-scale* experiments to effectively study interference patterns in multi-cell settings with high-power RF front-ends.
- Enabling *remote-controlled* operation and management of base stations in outdoor, permanent roof-top deployments.
- Being *cost-effective*. WARPv3 hardware was an expensive solutions, with per-radio costs of $2K or higher.

These objectives led us to building the third-generation of Argos platforms with software-defined radio hardware optimized for massive-MIMO, especially scalability in cost, management, form factors, and network operations [26]. ArgosV3 remains the largest massive-MIMO base station publicly known (160 radios). The successor to ArgosV3 is commercially available via Skylark Wireless and is part of the Platform for Advanced Wireless Research (PAWR) POWDER testbed [19] deployed on the campus of University of Utah in Salt Lake City.

(a) ArgosV1 (back view) [28]


(b) ArgosV2 (side view) [29].

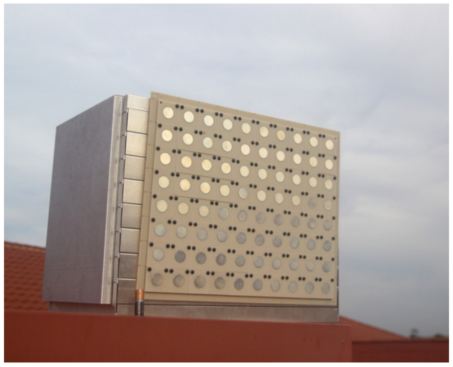

(c) ArgosV3 with AA battery for scale [26].

Figure 1: Three generations of Argos massive-MIMO base stations. Note the base stations are not shown to scale. ArgosV1 is approx. 1.1 x 1.25 x 0.6 m with 64 radios; ArgosV2 is approx. 0.5 m x 1.2 m x 0.35 m with 96 radios; and ArgosV3 is approx. 0.36 x 0.46 x 0.33 m with 160 radios – a reduction in almost 40-fold in volume per-radio across generations, and over 30-fold compared to other platforms.

2 BACKGROUND

MU-MIMO is a key technology that improves the spectral efficiency of wireless systems by employing many independent radios on the base station. The most practical technique for implementing MU-MIMO or Multi-User Beamforming (MUBF) is *linear precoding* where users' streams are precoded with a beamforming weight matrix. This matrix is calculated based on the channel state information (CSI) that is sent independently by each user to the base station. The base station, then forms the beamforming matrix W that, due to channel reciprocity, can be applied to both downlink and uplink. Since hardware gains and imperfection at both user and base station transceivers are not reciprocal, a fixed calibration matrix C is applied to uplink CSI for downlink beamforming. There are three predominant beamforming techniques: *i)* Conjugate, *ii)* Zeroforcing (ZF), and *iii)* MMSE. Both ZF and MMSE require an inversion of the CSI matrix, H, which is a centralized operation. For example, the ZF precoding matrix is: $W_{zf} = H^*(H^H H^*)^{-1}$. On the other hand, the conjugate method only needs the complex conjugate of the CSI matrix, i.e., $W_{conj} = H^*$. This can be done at each antenna without sending the entire CSI matrix to a central location.

ZF and MMSE are significantly more spectral efficient than conjugate method in high-SNR environments, however they incur a large processing and latency overhead due to their centralized nature. This can significantly degrade throughput when users are mobile; their channel state is constantly changing, thus the beamforming matrix not only has to be constantly recalculated, but can already be stale by the time it is used. In such scenarios, or in low-SNR environments, conjugate method can be more effective [22, 23].

3 DESIGN

Massive-MIMO technology requires scalability in terms of the number of radios in the base station. Work in [28] suggested a mix of "fat-tree" and "daisy chain" topologies and to interconnect basic building blocks both serially and in parallel, as shown in Fig 3a. Most existing massive-MIMO testbeds only implement the fat-tree topology shown in Fig. 3b, but do not scale easily to a high number of antennas.

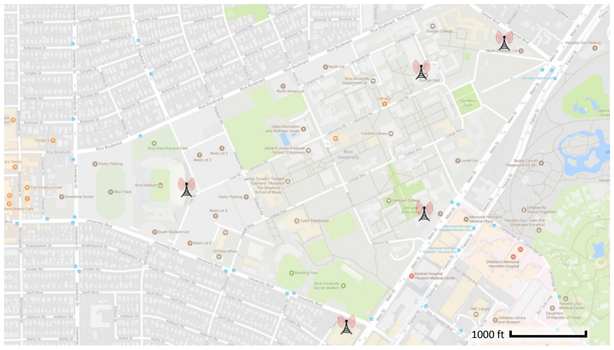


Figure 2: ArgosNet base station locations. Each location has mounting, power, and direct fiber to the server cluster. All locations are outdoor sites except for Duncan Hall *top middle*, which is our indoor lab environment [25].

Therefore, ArgosV3 adopts the architecture depicted in Fig. 3a for significantly better scalability and coordinated multicell support. For smaller scale standalone base stations, e.g., <160 radios, ArgosV3 implements Fig. 3c, which is a subset of Fig. 3a. In a daisy chain, it is important that not only data but clock and time synchronization triggers are also propagated serially, otherwise the intended scalability is not realized. However, each poses a unique challenge: quality of the clock deteriorates at each hop, when distributed serially; moreover, time synchronization triggers arrive with a delay to each radio in a daisy chain, which complicates synchronization. ArgosV3 addresses these challenges in an efficient and reliable way, as discussed in §4.

We next describe the design of ArgosNet, which consists of multiple ArgosV3 base stations, three deployed at rooftops of our campus, an on-campus server cluster, and mobile clients, as shown in Figs. 2, 7,and 8.

3.1 Building Block

The key design decision we made early on was to build a new SDR module, called *Iris*, specifically for scalable many-antenna base stations. In order to leverage spatial diversity gains, improve reliability, and reduce assembly labor, we focused on reducing both

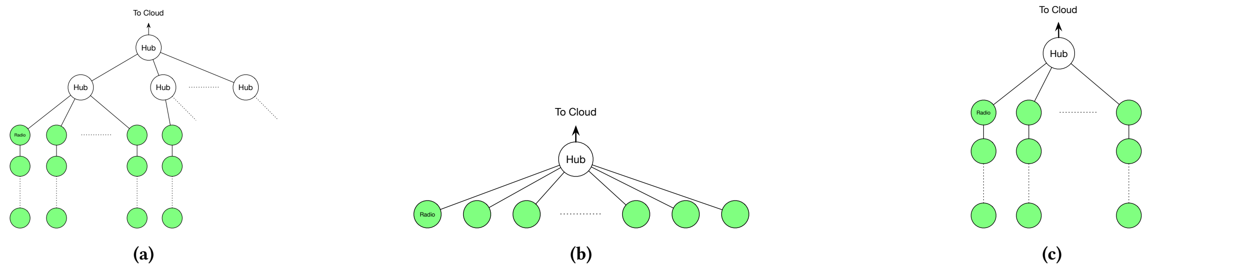


Figure 3: (a) The abstract Argos architecture as described in [28]; (b) Fat-tree topology actually implemented by most existing platforms; (c) Daisy-chained architecture implemented by ArgosV3. ArgosNet, consisting of multiple ArgosV3 base stations, actually implements the architecture shown in (a), with each hub corresponding to a base station and the central hub residing in the data center.

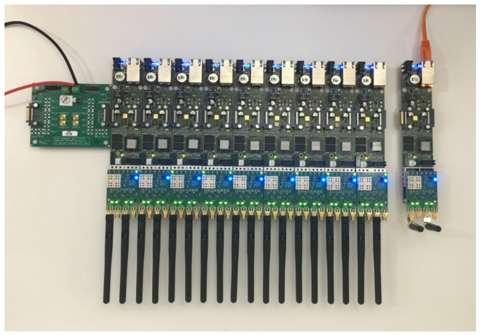


Figure 4: Array of 9 Irises demonstrating daisy-chain function and an individual Iris powered by PoE (*right*).

the unit form factor and the amount of cabling required compared to WARP, which was used in ArgosV1 and ArgosV2. A 4-radio WARP module was approximately 8 inches to a side while a comparable 4-radio Iris configuration is 3.1 in × 11.2 in. Moreover, for a 128-antenna ArgosV2 base station with 32 WARP modules, each module required four separate wired connections for power, clock, time synchronization trigger, and Ethernet to be pulled through separate cables from an egress point (see Fig. 1b). As a result, debugging, maintenance and reconfiguring the base station required significant effort, with each cable being a common point of failure. This is also true for other existing massive-MIMO testbeds.

Irises are highly optimized for massive-MIMO base stations as they can be daisy chained in a LEGO-like fashion, without any wiring, as shown in Fig. 4. Each Iris has two matching connectors on its sides, a male and a female, respectively. The connector includes pins for:

- A power bus so that the entire daisy chain is powered from the same source
- Differential clock lines from the master to the slave Iris
- Eight bi-directional GPIO lines
- A high-speed bi-directional data plane

Fig. 5 depicts the architecture of a daisy chain where the left-most Iris is connected to the *hub*. The hub provides 40 Gbps backhaul and acts as the gateway to the network. The Iris has a System-on-Chip (SoC) device, including FPGA fabric and an ARM processing unit running Ubuntu Linux used for running the control plane operations through device drivers. Additionally, each Iris implements an RF core for baseband processing required for MU-MIMO operations

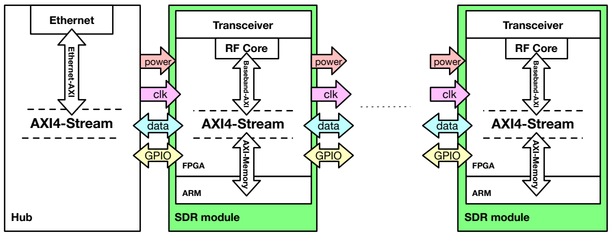


Figure 5: The block diagram of Iris modules daisy chained and connected to the Hub.

in the FPGA fabric. While the entire baseband processing can be run centrally in the cloud, there are usually latency and throughput constraints related to the wireless stack that necessitate offloading certain PHY operations to be as close to the radio as possible. The most typical of these processing blocks are the beamforming and FFT. Notably, these implementations in the RF core are transparent to the Ethernet-bearing AXI streams used in the rest of the design.

The data line is a high-speed serial full-duplex link between the FPGA fabrics of the two adjacent Iris modules where streams of samples or data are transmitted in both upstream and downstream directions. While the link itself does not use any particular standard, it can be used to transport any data associated to a data communication protocol, e.g., Ethernet, PCIe, CPRI, etc. In our solution, we used Ethernet frame format for data encapsulation in the FPGA for the two following reasons:

- To enable uniformity and reuse of logic on different components of the system without having to translate the stream at every junction point or at the hub (everything talks Ethernet, be it the Iris, the hub, or commodity switches and servers in the cloud)
- To simplify routing frames through the daisy chain with minimal FPGA logic.

We provide more details on this in §3.2.

3.2 Data Plane

The Internet, by nature, has unpredictable traffic; as such, it *cannot* guarantee packet delivery. This is mainly the result of unpredictable routing, and discrepant link speeds. ArgosNet heavily utilizes Ethernet encapsulation as the de-facto standard for its data plane to improve compatibility and flexibility while reducing cost and development time. However, it avoids Ethernet's unpredictability by: *i)* implementing data connections in the FPGA for deterministic

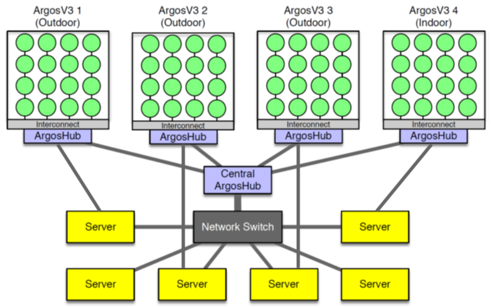


Figure 6: ArgosNet Architecture includes Argos base stations, Central ArgosHub, low-latency network switch and a pool of base-band processing servers, all interconnected through fiber.

performance, *ii)* routing consistent data paths, and *iii)* maintaining consistent and predictable data flows through appropriate link provisioning.

Inside the FPGA, data movement between blocks is enabled using the AXI4-stream data bus [8], which is a standard protocol for FPGAs and SoCs. Thus, every Ethernet frame arriving at the hub from the network is translated to AXI4 stream while maintaining the frame boundary through AXI4's flags. AXI4 streams pass through high-speed serial links on the daisy chain until they reach their destination Iris. However this requires each Iris to provide a routing mechanism. Thus, we implement a *stream router* block in each Iris's FPGA. The stream router snoops the MAC header information in the AXI4 stream and decides if it should route the stream to either: *i)* its own RF core if it contains baseband samples, *ii)* its memory to be handled by Linux on ARM as a control packet, or *iii)* downstream/upstream high speed serial links according to the destination Iris.

Network packets arriving at the hub are routed to their destination Iris. To facilitate this, the hub keeps a small routing table that maps each Iris's MAC to a destination daisy chain. Similarly, each Iris's traffic routed to the hub will be readily forwarded to the network through its backhaul connection.

3.3 Control Plane

ArgosNet includes various control plane functions that fall into two generic categories:

- Control operations necessary for efficient and flexible data movement between the base station and data center. This includes the configuration of Ethernet flows, originating from each Iris, by programmatically routing them to different destinations. It also considers functions at the Hub, for aggregation and dissemination of data between the Irises and the local server.
- Control functions related to the reconfiguration of wireless stack. These include radio transceiver settings such as gains and frequencies, the reconfiguration of the wireless TX/RX frames, control of FPGA state machines to enable bursty versus continuous streaming, etc.

The control plane follows a generic API and is implemented in a split stack architecture with *local* software module running on the local server and the *remote* module on the ARM processor at each Iris.

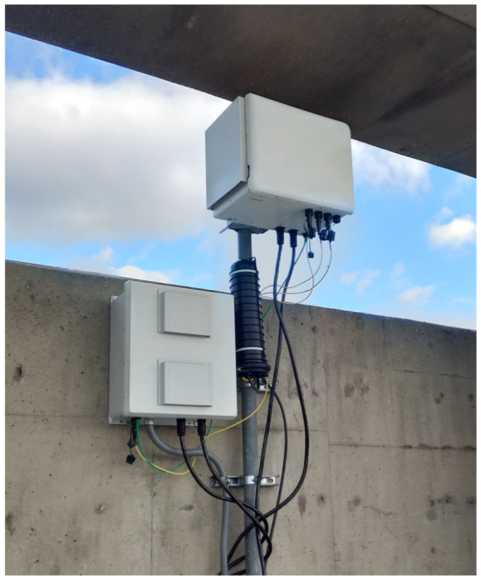


Figure 7: 160-radio weatherproof ArgosV3 base station deployed at Rice University stadium.

4 IMPLEMENTATION

ArgosNet, as shown in Fig. 6 is comprised of Argos base stations deployed both outdoors and indoors, as well as a central ArgosHub and a cloud server pool deployed at the data-center.

Each Argos base station is composed of two main components: *i)* Iris modules, and *ii)* one or more ArgosHubs.

Iris is a programmable SDR module and it is the main building block of Argos. It is based on Lime Mircosystem's LMS7002M [15] as the programmable radio transceiver and the Xilinx Zynq XC7Z030 (Z-7030) All Programmable SoC [9] as the processing element. LMS7002M is a frequency-agile 2×2 MIMO transceiver that can be tuned from 50 MHz to 3.8 GHz, sampling a maximum of 56 MHz of contiguous bandwidth on each radio interface with 12-bit ADC resolution. Xilinx Z-7030 allows running Linux on Iris, making it a very flexible system. Z-7030 includes four GTX high-speed serial transceivers in its programmable logic, that are each able to transfer data with rates up to 6.6 Gbps, in and out of the FPGA. Two GTX transceivers are routed to a connector on each side of a Iris for upstream and downstream communication, providing 13.2 Gbps in both directions. These connectors enable the daisy chaining capability that provides data connectivity, power sequencing, and time and frequency synchronization. Each Iris has an interchangeable dual-band 2.5GHz and 3.5GHz amplified RF front-end with a maximum 28 dBm output power. This enables a large coverage area and at-scale operation.

ArgosHub is a central aggregation unit at the base station which, through a high rate data bus, aggregates/distributes streams of samples from/to Iris modules in the base station. For additional scalability multiple hubs can be synchronized with fiber or direct coaxial connections to support even more radios on a single base station.

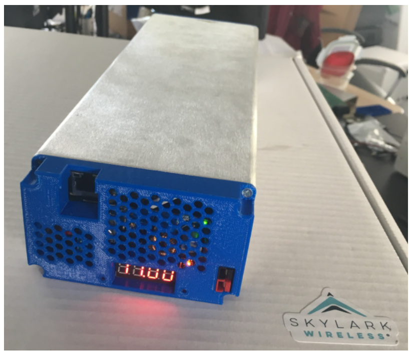


Figure 8: Battery powered ArgosV3 mobile client [25].

4.1 Clock Synchronization

In a massive-MIMO system, all radios must be locked to the same reference clock which is provided from an external source to all the radios. In Argos, the ArgosHub provides the reference clock to all Iris modules. While this is straightforward in a "fat-tree" topology, in a daisy chain topology, the reference clock has to propagate hop by hop to all the radios. This could potentially degrade the clock signal quality and introduce significant jitter or phase noise to the clock signal which in turn adversely affect channel coherence and beamforming operation [11].

To solve the issue of serial clock sharing, each Iris employs a reconfigurable ultra-low additive jitter clock buffer that de-jitters the reference clock received from the upstream, feeds it to its own RF chain, and sends it to the downstream clock buffer on the next hop.

To share a common clock among multiple ArgosV3 base stations, each ArgosHub implements Synchronous Ethernet (SyncE) where the clock is recovered from the Ethernet or fiber line. As shown in Fig. 6, there are direct fiber lines from the central ArgosHub to each ArgosV3 base station, which not only carry data but also a reference clock. In this setup, the central ArgosHub, or a SyncE switch, provides the master clock reference. Alternatively, the ArgosHub can also take a direct coaxial input for a reference clock and time-sync, e.g., to support GPS or connections to other co-located hubs.

4.2 Time Synchronization

ArgosV2 enabled time synchronization by sending pulse triggers from a master node to all other nodes through GPIO lines [29]. While this simple solution works for ArgosV2, it is not sufficient for daisy chain synchronization. For time synchronization to work with a daisy chain topology, we have to compensate for the propagation delay along each hop of the chain. Thus every Iris must know the delay it take the trigger signal to arrive at the previous Iris module. To measure this delay we undertake the following procedure using a SYNC signal.

(1) A SYNC signal is sent to a chain from a master Iris or the ArgosHub, through dedicated lines (serial, Ethernet, etc.).
(2) Each Iris, upon receiving the SYNC signal, starts a DELAY counter while forwarding the signal down the chain.
(3) Once the last Iris receives the SYNC signal, it sends it back to the chain in the reverse direction.

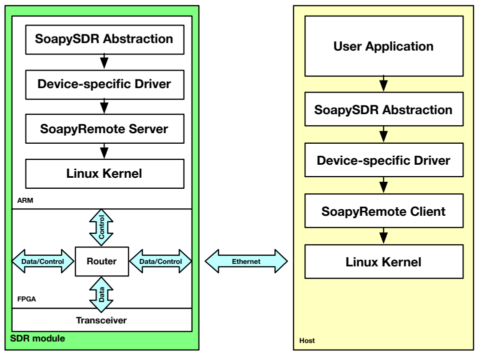


Figure 9: SoapySDR flow and Iris Software Driver in a split-stack architecture, with the data and control plane separated.

(4) Each Iris, once receives the looped back SYNC signal, locks its DELAY counter value.
(5) Each Iris uses the DELAY/2 value to internally delay any trigger command it receives.

This procedure is available through a control plane API function call from software and only done once at the beginning of the operation.

A similar procedure is used to synchronize multiple ArgosHub modules using the central ArgosHub for multi-cell time synchronization.

4.3 Software Framework

To enable rapid development and prototyping with ArgosNet, we leveraged the open-source *SoapySDR* framework [5], a project aimed at providing common API that supports various SDR devices. Specifically, it provides a set of abstract functions for radio configuration and streaming that is common among many SDR platforms, e.g., USRP[20]. Any SoapySDR-based application interacts with these SDRs through device-specific implementations wrapped as shared libraries, which provides excellent portability between SDR platforms. While the API is written in C++, it provides Python bindings for rapid prototyping and development.

Another important part of the Soapy Ecosystem is the *SoapyRemote* framework that provides a TCP/IP-based API for communicating with remote devices. It implements both server and client entities with the server running on the remote device and the client in the host. It also wraps all SoapySDR functions calls as RPC calls from client to the server.

Fig. 9 shows the software framework architecture implemented for Argos where each Iris is controlled independently through a split-stack architecture with separate data and control plane.

5 EVALUATION

To evaluate the system, we implemented a channel measurement framework using SoapySDR API, based on the system presented in [24]. In this framework, all Irises in the Argos base station receive pilot training signals that are sent orthogonally in time by an arbitrary number of standalone Irises emulating mobile clients. It is known that in a static channel scenario, i.e., when the mobile clients, objects and people in the environment are not moving, channel coherence time is long and the pilot signals remain highly

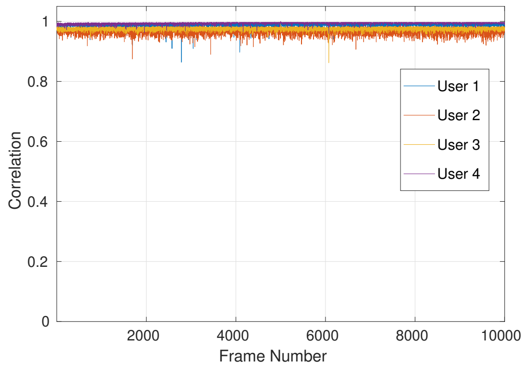


Figure 10: Temporal correlation of uplink pilot signals collected from 4 users on a 64-antenna Argos base station in a static channel scenario over 50 s with 5 ms frames.

correlated [23]. To to be able to verify this behavior in a massive-MIMO channel, all radios in the system must be phase-locked and remain time-synchronized during the measurement. Fig. 10 shows this temporal correlation in pilots collected from four users during a 50-second long experiment (divided into 5 ms frames) by a 64-antenna Argos base station (including four daisy-chains each with 8 Iris modules). It is observed the correlation line is flat and always above 95% which proves the effectiveness of the Argos design.

6 RELATED WORK

Commercially available massive-MIMO testbeds: To the best of our knowledge, there are two commercially available massive-MIMO testbeds reported by the industry, including a platform from National Instruments [16] and TitanMIMO [18] from Nutaq. Among them, NI testbed is probably the most widely used in the community used by several academic institutions for massive-MIMO research [1, 7, 30].

NI platform interconnects up to 64 USRP devices through a master PXI chassis to form a 128-antenna base station. Due to heavy use of PCIe standard, it cannot be scaled beyond 128-antennas. Moreover, due to its large footprint and limited transmit power it is only suitable for lab testing environments. Additionally, it does not provide an easy integration path for Cloud RAN or Coordinated Multi-Point (CoMP) operations.

TitanMIMO uses Perseus boards with 4 or 8 radio transceivers as its building block. Each board provides high-bandwidth backhaul with flexibility to interconnect with a user-defined topology. However this indicates a tailored baseband design for each toplogy. TitanMIMO also lacks any amplified analog RF front-end, and therefore it cannot be considered as a deployment-ready massive-MIMO platform.

Deployment and Commercial Offerings: Verizon and Ericsson announced deployment of a 96-antenna massive-MIMO base station with 16 digitally beamformed chains in 2017 [6]. Later, in the same year, the same team together with Qualcomm tested FDD massive-MIMO using the latest 3GPP release LTE supporting 8-user terminals [3].

Samsung and Sprint partnered to deploy Samsung's FD-MIMO technology [2]. Their test included a 32-antenna base station and demonstrated MU-MIMO with 8 user terminals. While massive-MIMO service will likely be offered by operators soon, its scale of deployment, size of the arrays, and flexibility is still far from what can be achieved with ArgosNet.

Academic Attempts: Eurecom's massive-MIMO testbed is mainly built to support OpenAirInterface 4G-LTE stack [10]. It uses a proprietary EXMIMO2 SDR as its building block with 4 transceiver chains and a Gen-1 PCIe interface. Through a PCIe chassis, this platform interconnects up to 16 EXMIMO2 modules, creating a 64-antenna base station. Leveraging the same PCIe chassis, the radios communicate with a host PC that runs baseband processing centrally. This platform has been mainly built to support OpenAirInterface 4G-LTE stack and it does not support backhaul for Cloud-RAN or synchronized multi-cell operations.

7 CONCLUDING REMARKS

We designed and built ArgosNet from the board-level up to be a flexible massive-MIMO platform that enables at-scale deployment in realistic environments. Drawing on our experience in building two previous generations of massive-MIMO systems, ArgosNet provides unprecedented scalability and programmability, as well as enables operation in real-world outdoor multi-cell topologies.

However, as expected, designing and implementing such a complex multi-cell massive-MIMO platform presented numerous challenges. Beyond the architectural and synchronization hurdles discussed above, building and producing custom hardware is always a time-consuming and expensive task. The project took over 3 years of testing and iteration, with many rounds of prototypes and mechanical designs. Moreover, manufacturing in volume and ensuring functionality is a huge task in itself, as it requires careful testing and tracking – the testing hardware and software alone was a challenging project that required multiple custom board designs. With 7 custom PCB designs and over 600 PCBs deployed in ArgosNet, proper volume manufacturing was paramount.

While massive-MIMO technology is considered to be a key technology for 5G, commercial offerings of many-antenna base stations rarely have more than 64 antennas, and do not support more than 8 spatial streams. Importantly, many of these systems do not support fully-digital massive-MIMO. That is, they feature a hybrid design where there are far fewer transceivers than antennas, by using analog phased arrays. For example, Verizon and Ericsson recently announced a 96-antenna base station with only 16 active transceivers [6]. Similarly, Samsung's FD-MIMO [12] product features only 32 transceivers. This suggests the scalability and economics of commercial massive-MIMO systems is still a significant and pressing issue. Furthermore, based on our understanding, hardware and energy cost are critical industry concerns.

ArgosNet demonstrates the feasibility and performance of at-scale multi-cell massive-MIMO *networks*, enabling next-generation research and innovation in realistic environments.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants EARS 1444056, CRI 1405937, CNS 1518916, SBIR 1520496, and SBIR 1632565. We thank Lev Shuhatovich, Bryan Grandy, William Deigaard, Ashutosh Sabharwal, and Edward Knightly for their input, support, and help.

REFERENCES

[1] 2017. Facebook's Massive New Antennas Can Beam Internet for Miles. https://www.wired.com/2016/04/facebooks-massive-new-antennas-can-beam-internet-miles/.

[2] 2017. *Massive MIMO comes of age.* Technical Report. Ovum. http://www.samsung.com/global/business-images/insights/2017/Massive-MIMO-Comes-of-Age-0.pdf

[3] 2017. Verizon, Ericsson, Qualcomm Demonstrate FDD Massive MIMO. https://www.microwavejournal.com/articles/29576-verizon-ericsson-qualcomm-demonstrate-fdd-massive-mimo.

[4] E. Blossom. 2004. GNU radio: tools for exploring the radio frequency spectrum. *Linux Journal* 2004, 122 (June 2004).

[5] J. Blum. 2018. SoapySDR Project. https://www.ettus.com/sdr-software/detail/usrp-hardware-driver.

[6] Erricson. [n.d.]. Verizon and Ericsson team up to deploy Massive MIMO. https://www.ericsson.com/en/press-releases/2017/10/verizon-and-ericsson-team-up-to-deploy-massive-mimo.

[7] P. Harris, W. B. Hasan, H. Brice, B. Chitambira, M. Beach, E. Mellios, A. Nix, S. Armour, and A. Doufexi. 2016. An overview of massive MIMO research at the University of Bristol. In *Radio Propagation and Technologies for 5G (2016)*. 1–5.

[8] Xilinx Inc. 2011. AXI Reference Guide. https://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf.

[9] Xilinx Inc. 2017. Zynq-7000 All Programmable SoC Data Sheet: Overview. https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf.

[10] X. Jiang and F. Klatenberger. 2017. Demo: an LTE compatible massive MIMO testbed based on OpenAirInterface. In *International ITG Workshop on Smart Antennas*.

[11] R. Krishnan, M. R. Khanzadi, N. Krishnan, Y. Wu, A. G. Amat, T. Errikson, and Schober R. 2016. Linear Massive MIMO Precoders in the Presence of Phase Noise – A Large-Scale Analysis. *IEEE Transactions on Vehicular Technology* 65, 5 (May 2016), 3057–3071.

[12] Yang Li, Yan Xin, Mian Dong, G. Xu, J. C. Zhang, Younsun Kim, and Juho Lee. 2013. Implementation of full-dimensional MIMO (FD-MIMO) in LTE. In *Proc. Asilomar Conf. Signals, Systems and Computers (ACSSC)*. 998–1003.

[13] T.L. Marzetta. 2006. How Much Training is Required for Multiuser MIMO?. In *Proc. IEEE Asilomar Conf. on Signals, Systems and Computers (ACSSC)*.

[14] T.L. Marzetta. 2010. Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas. *IEEE Trans. on Wireless Communications* (2010).

[15] Lime Microsystems. 2017. LMS7002M DataSheet. http://www.limemicro.com/wp-content/uploads/2017/07/LMS7002M-Data-Sheet-v3.1r00.pdf.

[16] National Instruments Inc. [n.d.]. Introduction to the NI MIMO Prototyping System Hardware, Dec. 2017. http://www.ni.com/white-paper/53197/en/.

[17] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet. 2014. OpenAirInterface: A Flexible Platform for 5G Research. In *ACM SIGCOMM Computer Communication Review*. ACM, 33–38.

[18] Nutaq Inc. [n.d.]. 5G massive MIMO testbed. https://www.nutaq.com/5g-massive-mimo-testbed.

[19] Platform for Open Wireless Data-driven Experimental Research. [n.d.]. https://www.powderwireless.net/.

[20] Ettus Research. 2018. UHD (USRP Hardware Driver™). https://www.ettus.com/sdr-software/detail/usrp-hardware-driver.

[21] Rice University Wireless Open Access Research Platform. [n.d.]. warp.rice.edu.

[22] Clayton Shepard, Narendra Anand, and Lin Zhong. 2013. Practical Performance of MU-MIMO Precoding in Many-antenna Base Stations. In *Proc. ACM CellNet Workshop*.

[23] Clayton Shepard, Jian Ding, Ryan E Guerra, and Lin Zhong. 2016. Understanding real many-antenna MU-MIMO channels. In *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE, 461–467.

[24] Clayton Shepard, Jian Ding, Ryan E Guerra, and Lin Zhong. 2016. Understanding real many-antenna MU-MIMO channels. In *50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 461–467.

[25] C. Shepard, R. Doost-Mohammady, J. Ding, R. E. Guerra, and L. Zhong. 2018. ArgosNet: A Multi-Cell Many-Antenna MU-MIMO Platform. In *52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2237–2241.

[26] Clayton Shepard, Rahman Doost-Mohammady, Ryan E. Guerra, and Lin Zhong. 2017. ArgosV3: An Efficient Many-Antenna Platform. In *Extended Demonstration Abstract in Proc. ACM MobiCom*.

[27] Clayton Shepard, Abeer Javed, and Lin Zhong. 2015. Control Channel Design for Many-Antenna MU-MIMO. In *Proc. ACM MobiCom*.

[28] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong. 2012. Argos: Practical Many-Antenna Base Stations. In *Proc. ACM MobiCom*.

[29] Clayton Shepard, Hang Yu, and Lin Zhong. 2013. ArgosV2: A Flexible Many-Antenna Research Platform. In *Extended Demonstration Abstract in Proc. ACM MobiCom*.

[30] Joao Vieira, Steffen Malkowsky, Karl Nieman, Zachary Miers, Nikhil Kundargi, Liang Liu, Ian Wong, Viktor Owall, Ove Edfors, and Fredrik Tufvesson. 2014. A flexible 100-antenna testbed for Massive MIMO. In *IEEE GLOBECOM Wrkshp. Massive MIMO: from theory to practice*.