



Jacob Holtom *D, Andrew Herschfelt *, Isabella Lenz D, Owen Ma, Hanguang Yu and Daniel W. Bliss

The Center for Wireless Information Systems and Computational Architectures (WISCA), Arizona State University, Tempe, AZ 85281, USA

* Correspondence: jholtom@asu.edu (J.H.); andrew.herschfelt@asu.edu (A.H.)

Abstract: Validating RF applications is traditionally time consuming, even for relatively simple systems. We developed the WISCA Software-Defined Radio Network (WISCANet) to accelerate the implementation and validation of radio applications over-the-air (OTA). WISCANet is a hardwareagnostic control software that automatically configures and controls a software-defined radio (SDR) network. By abstracting the hardware controls away from the user, WISCANet allows a non-expert user to deploy an OTA application by simply defining a baseband processing chain in a high level language. This technology reduces transition time between system design and OTA deployment, accelerates debugging and validation processes, and makes OTA experimentation more accessible to users that are not radio hardware experts. WISCANet emulates real-time RF operations, enabling users to perform real-time experiments without the typical restrictions on processing speed and hardware capabilities. WISCANet also supports multiple RF front-ends (RFFEs) per compute node, allowing sub-6 and mmWave systems to coexist on the same node. This coexistence enables simultaneous baseband processing that simplifies and enhances advanced algorithms and beyond-5G applications. In this study, we highlight the capabilities of WISCANet in several sub-6 and mmWave over-the-air demonstrations. The open source release of this software may be found on the WISCA GitHub page.

Keywords: software-defined radio; software-defined radio network; cognitive radio; multiple-input multiple-output; integrated sensing and communications; spectral convergence; sub-6; mmWave

1. Introduction

The fifth generation (5G) of wireless communications systems has quickly become a global industry standard. To address the growing list of requirements not satisfied by the existing 5G New Radio (NR) standards, academia and industry have shifted their focus to beyond-5G (B5G) and 6G systems. Several reputable agencies have established organizations to address this problem, including: *Technologies for Network 2030* initiated by the International Telecommunication Union (ITU) in July 2018 [1]; *6G Vision and Drivers* launched by Generation Mobile Networks (NGMN) in 2020 [2]; and a group of projects referred to as ICT-20 *5G Long Term Evolution* and ICT-52 *Smart Connectivity beyond 5G* initiated by the European Commission in 2020. The United States, China, Germany, Japan, and South Korea have all published plans to pursue 6G research [3].

To augment the capabilities of 5G systems, multi-function systems simultaneously integrate multiple wireless functionalities, including sensing, communications, computations, and controls [4–7]. These hybrid technologies enable mutual performance enhancements and novel capabilities, but also demand higher computational complexity and more capable RF front-ends (RFFEs) [8,9]. Several emerging solutions significantly mitigate these hardware and software limitations, including domain-specific heterogeneous processors [10–12]; multi-access techniques and hybrid processing architectures [13–16]; machine learning (ML)



Citation: Holtom, J.; Herschfelt, A.; Lenz, I.; Ma, O.; Yu, H.; Bliss, D.W. WISCANet: A Rapid Development Platform for Beyond 5G and 6G Radio System Prototyping. *Signals* 2022, *3*, 682–707. https://doi.org/ 10.3390/signals3040041

Academic Editors: Bo Yang, Hongliang Zhang and Shuping Dang

Received: 1 August 2022 Accepted: 20 September 2022 Published: 9 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). and artificial intelligence (AI) techniques [17–21]; and reconfigurable intelligent surfaces (RISs) [22–24].

Despite the incredible capabilities of these emerging technologies, implementing and validating functional prototypes is still a tedious and time-consuming challenge. SDRs [25–27] and RF system-on-chips (RFSoCs) [28,29] have greatly simplified this process, but still require some expertise to use efficiently.

To accelerate the implementation and validation of numerous B5G/6G applications, we developed the WISCA Software-Defined Radio Network (WISCANet) [27,30]. WIS-CANet is a hardware-agnostic control software that automatically configures and controls a software-defined radio (SDR) network. By abstracting the hardware controls away from the user, WISCANet allows a non-expert user to deploy an over-the-air (OTA) application by simply defining a baseband processing chain in a high level language. WISCANet emulates real-time operations, allowing users to perform real-time experiments without the typical restrictions on processing speed and hardware capabilities. By abstracting the hardware controls and emulating real-time operations, WISCANet allows end-users to quickly deploy radio applications without becoming hardware experts or rigorously optimizing their software. To further improve portability, accessibility, and ease-of-use, we continue to expand the supported hardware and programming languages (MATLAB, Python, C/C++, Rust, Julia, etc.). This architecture is depicted in Figure 1.

To support higher frequency ranges, WISCANet deployments may be augmented by either mixing directly from baseband or superheterodyning inside the equipped RFFEs. In our specific deployment (pictured in Figure 2), we have executed over-the-air applications as high as 71 GHz. To support 6G research, we plan to extend support into the THz band; unfortunately, commercial THz SDRs are not widely available, so we are actively pursuing a superheterodyning approach similar to the technique discussed later in this paper. Other researchers have successfully built small-scale THz test systems using the same strategy [31].



Figure 1. The WISCANet control software runs on a control computer, which receives a user-defined application and automatically configures the connected edge nodes and radio to execute it. This allows users to deploy over-the-air radio applications without programming the radios themselves, which reduces development time and makes experimentation more accessible to beginners.

1.1. WISCANet and B5G/6G

The primary objective of the WISCANet SDR testbed is to accelerate the development cycle of RF applications by reducing the cost of early implementation and validation stages. Many emerging B5G/6G technologies are algorithmic innovations to existing hardware; WISCANet allows algorithm experts to quickly test their new ideas without the hassle of building and operating custom hardware. WISCANet's features specifically enable many of the B5G/6G techniques identified in this special issue, some of which are enumerated below.

Size and Scalability

WISCANet supports dozens of simultaneous radio devices, allowing a user to quickly configure and deploy OTA network applications without the usual overhead of actually building a network. This enables rapid OTA validation of novel network techniques that are typically only characterized in simulation. Emerging B5G/6G network techniques in this area include wireless resource allocation; mobility management; multi-access; internet-of-things (IOT); spectrum sensing and sharing; and distributed coherence.

Flexibility

WISCANet allows users to quickly reprogram radios for a variety of different tasks, enabling a range of customizable mission scenarios for characterizing channel estimation and predicition; interference avoidance, management, and cancellation; and localization.

Programmability

WISCANet automatically configures the connected radios with minimal user input, so end users can easily test a variety of adaptive techniques that modify the processing chains in real-time, including dynamic spectrum sensing and access; adaptive waveforms and fluid protocols; and novel modulation techniques.

Hardware Integration

WISCANet easily integrates with hardware accelerators like CUDA GPUs, FPGAs, and RFSoCs to accelerate computation-heavy applications. These accelerators have proven extremely useful in enabling a variety of machine learning (ML) and artificial intelligence (AI) techniques [32–35].

Compatibility

WISCANet is compatible with the industry-standard Ettus USRP SDRs and several programming languages (MATLAB, Python, C/C++, Rust, etc.), making it accessible and budget-friendly for academic research institutions or independent developers. This further facilitates portability and interoperability between different devices, enabling a variety of IoT applications.

Upgradability

WISCANet can be augmented with high-performance equipment like OctoClock distribution modules to enable precise synchronization and time control for sensitive applications such as ultra-reliable, low-latency communications; precise localization; and distributed coherence.

Stability

WISCANet features phase synchronization and stability features between connected radios, enabling massive multiple-input, multiple-output (MIMO) applications and numerous multi-antenna techniques.

1.2. Contributions

In this paper, we make the following contributions:

- Describe recent updates to the WISCANet testbed.
- Demonstrate several sub-6 and mmWave OTA applications on the WISCANet testbed, including:
 - mmWave Communications;
 - 5G positioning and timing;
 - distributed mosaic beamforming;
 - adaptive waveform deployment; and
 - dynamic spectrum access.
- Release an updated version of WISCANet under the Lesser GNU Public License Version 3.0 (LGPLv3.0).
- Release example applications for WISCANet under the LGPLv3.0.

1.3. Organization

This manuscript is organized as follows: in Section 2, we survey the existing literature on emerging 5G technologies and similar SDR testbeds; in Section 3, we describe the system architecture and features of WISCANet; and in Section 4, we demonstrate several sub-6 and mmWave applications over-the-air.



Figure 2. Example deployment of the WISCANet control architecture on several Ettus USRP devices controlled by Dell workstations and Intel NUC Small Form Factor PCs (SFFPCs). This deployment features 8 Ettus B210s and 5 Ettus X310s which operate in the typical sub-6 frequency bands. This implementation also features (2) 60 GHz phased array antennas for mmWave experiments. This deployment can be used for a variety of experiments in both mmWave and sub-6 GHz bands, enabling diverse, multi-band, multi-antenna experiments but also remaining accessible to users without advanced networking and programming expertise.

2. Background

In this section, we briefly survey recent advances in B5G/6G technologies, softwaredefined radios, and related SDR testbeds developed by other organizations. The WISCANet software architecture was specifically motivated by the accessibility of Ettus SDR devices and similar SDR network testbeds described in Section 2.3. We further compare WISCANet to traditional approaches in Section 2.5 to highlight the advantages and acknowledge the disadvantages of our solution.

2.1. Recent Advances in B5G/6G Technologies

Numerous emerging technologies have begun to address the shortcomings of existing 5G systems. Novel management architectures [36] and machine learning techniques [37] have begun to address challenges in mobility management and wireless resource allocation. Wireless power transfer [38] and energy harvesting [39] are becoming viable approaches for satisfying distributed power supply needs. Channel estimation and prediction have been better characterized [40] and improved by deep learning techniques [41]. New modulation and multi-access techniques address network congestion and improve mutual performance [42,43] while compressed sensing is improving source coding for challenging environments [44,45]. Internet-of-things applications are becoming more viable and capable [46]. Reconfigurable intelligent surfaces (RISs) can help overcome channel loss associated with high frequency bands and can be exploited for passive beamforming [3]. Visible light communication (VLC), optical networks, and aerial access networks can support very high data rates and expand coverage area [21,47].

2.2. Software-Defined Radios

Software-defined radios (SDRs) have enabled implementation and validation of RF systems without specialized hardware or advanced programming skills. Growing support from the open source community continues to support these increasingly accessible and affordable technologies. We briefly summarize the key SDR technologies that enable the advanced features offered by WISCANet.

2.2.1. Ettus USRP and UHD

The Universal Software Radio Peripheral (USRP) is a product series developed by Ettus Research [48]. This product line consists of embedded system-on-chip (SOC) radios (E series); Universal Serial Bus (USB) radios (B series); Ethernet radios (N series); and wide bandwidth radios (X series). Each model targets different frequency ranges, sampling rates, and processing requirements [49]. The USRP Hardware Driver (UHD) is an open-source C++ library and API for systems leveraging the USRP, which can be used to build radio applications without understanding the full details of the underlying hardware [50]. Conveniently, MATLAB provides a hardware support package that integrates the UHD and associated firmware into the MATLAB development platform [51].

2.2.2. GNURadio

GNURadio is a free and open-source software radio ecosystem. The project is written and developed for both C++ and Python [52,53]. GNURadio has a built-in block library that contains a number of useful signal processing components and provides a Graphical User Interface (GUI) to connect these components into a processing chain—called a flowgraph for real-time execution [54]. GNURadio greatly simplifies the implementation of simple radio applications, but is primarily limited by how many SDRs it can manage with a single host computer. Developing custom processing blocks in GNURadio can often be tedious, so it is typically limited to single-radio applications. GNURadio performs excellently in this use-case but does not provide the capabilities required for larger network applications.

687

2.2.3. SoapySDR

SoapySDR is a general purpose library for interacting with a variety of SDR devices [55]. It enables researchers and developers to build software that operates across many different SDR platforms using the same functional API. SoapySDR is part of the open-source Pothosware SDR ecosystem [56], which provides SDR development tools including a GUI similar to GNURadio Companion.

2.3. Related SDR Networks

WISCANet was inspired by several related SDR networks developed at various research institutions. These are briefly summarized below. Reconfigurable Ecosystem for Next-generation End-to-end Wireless (RENEW) and its companion software RENEWLab is a software suite that operates with the Faros and Iris hardware [57] to develop applications for experimentation on the RENEW platform [58,59]. WARPLab is a framework designed for rapid Physical Layer (PHY) prototyping [60,61]. It utilizes MATLAB and FPGA implementations of the framework and runs on its own reference hardware implementation rather than utilizing common off the shelf (COTS) SDR hardware. Other SDR networks have been proposed or implemented in [62–66]. In those works, customized or commercial SDR devices are used to suit the authors specific needs and research applications. In [67], the authors develop a system similar to WISCANet but instead use GNURadio as the intermediary between MATLAB and UHD. WISCANet provides a more flexible interface and a more efficient method of interfacing with USRPs using MATLAB, Python or other high level languages.

2.4. mmWave Testbeds and Frameworks

New mmWave technologies are often tested using expensive, custom-built experimental prototypes; as a result, most academic mmWave research is limited to simulation [68]. Several emerging solutions promise a lower-cost, open-source alternative to make mmWave experimentation more accessible. In [69], a reconfigurable MIMO-SAR testbed is presented using COTS mmWave sensors and motorized rail platforms. This solution is reconfigurable but has only been implemented for SAR applications. In [68], an affordable, SDR-based testbed called *MillimeTera* was proposed but not yet implemented. Despite the established need for mmWave sensing and communication testbeds, there are only a few accessible solutions.

2.5. WISCANet vs. Traditional Approaches

Commercial radio products are typically implemented on either FPGAs or ASICs. These implementations offer high efficiency and performance, but are also time-consuming to develop, require specialized hardware knowledge, and typically incur significant one-time engineering costs. These embedded solutions are excellent for mass-produced technologies, but are neither viable nor accessible for conducting research on novel RF techniques. Commercial SDRs have made entry-level RF research more accessible and affordable, but often do not scale well to network applications or high-precision techniques. While the open source community has developed excellent tools to support these devices, these solutions are typically insufficient for intermediate to advanced RF research.

WISCANet aims to address these shortcomings by providing a scalable control architecture for commercial SDR products. WISCANet allows users to quickly assemble SDR networks without requiring specialized hardware knowledge, and test over-the-air applications with minimal hardware configuration. By abstracting the hardware controls away from the user, WISCANet allows a non-expert user to deploy an OTA application by simply defining a baseband processing chain in a high level language. This technology reduces transition time between system design and OTA deployment, accelerates debugging and validation processes, and makes OTA experimentation more accessible to users that are not radio hardware experts.

3. System Design

WISCANet is a hardware-agnostic control software that automatically configures and controls an SDR network. A complete WISCANet deployment consists of a control node that runs this software, some number of edge computing nodes that receive instructions from this control node, and radio devices (SDRs, USRPs, RFFEs, etc.) connected to these edge nodes that perform RF operations. This configuration is depicted in Figure 1. The control node takes a user-defined baseband processing application and handles application distribution, hardware configuration, and edge node execution to deploy this application over-the-air. This control software is implemented in C++ using the C++17 standard, leverages the Secure Shell (SSH) protocol for edge node provisioning, and performs management over a secondary Transmission Control Protocol (TCP) channel. The system is configured in a set of files written in the industry standard YAML Ai not Markup Language (YAML), an example of which is provided in Listing 1. These files configure the edge node application paths, operating modes, frequencies, sampling rates, device addressing and more.

Each edge node interfaces with radio devices using the USRP Control subprocess. When instructed to, the edge node launches the user-defined baseband application at the runtime specified by the control node and interfaces this application with the control process. An example of this interaction is depicted in Figure 3 for an Ettus USRP peripheral. At the start of each application, the system initializes all edge nodes to an initial start time and maintains stable phase between channels. After this initial start time, transmit and receive timing is defined by the user software, as depicted in Figure 4.



Figure 3. Example interactions between a control node, edge node, and Ettus radio peripheral. The control node performs command and control operations over the TCP Control Channel and distributes baseband software over the SSH protocol. The edge node software has multiple threads, one controlling attached USRPs and the other running the user's application. They communicate through WISCANet bindings that leverage UDP for inter-process communication.

WISCANet has a stable Application Binary Interface (ABI), including an RF sample line protocol over the User Datagram Protocol (UDP), which is designed to be programminglanguage agnostic. This allows users to define applications in any language and—when run inside the WISCANet framework—gain access to WISCANet's quasi-real-time RF transmit and receive capabilities and diverse feature set. We already provide bindings for MATLAB, Python, and GNURadio, and can easily extend support to other languages. WISCANet supports Ettus USRP radios that support the UHD interface. This enables support for flexible multi-antenna and multi-channel operation, including phasesynchronous operation. WISCANet requires GPS synchronization between all Ettus radio frontends; however, more complex networks may utilize the UHD multi_usrp interface, which enables multiple USRPs to behave as one radio unit by configuring them to share a single reference GPS, 10 MHz clock, and pulse-per second signal. This, in turn, enables phase synchronized operation across multiple channels, including stable phase between on-air frames in the quasi-real-time system. This enables beamforming and MIMO applications leveraging estimates made in prior cycles. The Octoclock-G may also be leveraged to provide synchronization to radios in the network. When different models of USRP are used in a single system, there must be some small compensation for the differing baseband delay in the time to first sample. These are fixed quantities determined by the different FPGA routing delays in different radio firmware. If the on-air time is sufficiently long and there is enough guard time, this tends toward being a negligible difference.

Listing 1. Example edge node YAML configuration file. This file configures a simple 4-channel communications application using an Ettus X310, MATLAB runtime, and 10 MHz sampling rate at a center frequency of 907 MHz.

```
1
2
    logic_id: 1 # Unique numeric identifier for each edge node
    op_mode: TX/RX # Operating Mode - TX, RX, or TX/RX
3
    mac_mode: UMAC # MAC Mode - UMAC (User Defined MAC - dynamically configurable time steps)
4
5
    time_slot: 1 # TDMA MAC Mode only
6
    lang: matlab # Runtime language (matlab, python, gnuradio, etc...)
    matlab_dir: CommunicationsAPI/USRP # Subfolder for executable script
7
    matlab_func: usrpPingPong # Program Name
8
    matlab_log: NULL
    num_samples: 50000 # Number of samples per cycle
10
    sample_rate: 10e6 # Sample rate
11
    subdevice: "A:A A:B" # UHD Subdevice
12
    freq: 907e6 # Center Frequency
13
    tx_gain: 20 # Transmit Gain
14
    rx_gain: 20 # Receive Gain
15
    bandwidth: 100E6 # USRP Frontend Filter Bandwidth
16
    device_addr: addr0=192.168.30.2, addr1=192.168.40.2 # UHD Device Address
17
    channels: "0,1,2,3" # UHD Channel Specification
18
    antennas: "TX/RX" # UHD Antenna Specificaiton
19
```



Figure 4. WISCANet emulates real-time operations by executing a single RF frame, freezing all nodes' transmit and receive operations, performing baseband processing in MATLAB (or other compatible language), and resuming all RF operations on the edge nodes at the same time for the next frame. From the radios' perspectives, this application looks real-time; but the extra processing time allows users to test minimally-viable code without needing to optimize and accelerate it (traditionally a tedious and time-consuming process). This also allows WISCANet to support languages like MATLAB that are traditionally slower than more optimal languages.

WISCANet provides a simple, consistent interface to the developer in any of the supported languages and bindings. The first step is to initialize the LocalUSRP object. The baseband software is provided an initial start time as a runtime argument or parameter. Afterward, the baseband software developer can call the tx_usrp and rx_usrp member functions, specifying the start time and number of channels desired. The key function signatures are as follows:

```
rx_buffer = rx_usrp(start_time, num_chans);
tx_usrp(tx_buffer, start_time, num_chans, reference_power);
```

where the buffer's will be complex double matrices of $(N_{samples} \times M_{channels})$ containing time-domain I/Q samples. The system expects them to be scaled into [-1, 1].

3.2. Phase Synchronization and Stability

To demonstrate WISCANet's phase-stable behavior between cycles, a sinusoidal tone is transmitted at a fixed carrier frequency into a power divider, which then feeds through phase-matched, equal length cables to each channel of a WISCANet configured phase-synchronized receiver. A block diagram of this experiment is depicted in Figure 5. The incoming phases of the reception at each channel of the receiver are compared. Results of these comparisons [27] are shown in Figures 6 and 7. In Figure 6, we observe that throughout each system transmit/receive cycle, stable phase is maintained. At the cycle boundary some number of 2π jumps occur. The same comparison plot is performed in Figure 7, for a 4 channel receiver, between the 6 channel pairs to observe the same synchronized behavior. Throught a cycle the relative phase stable, and the relative phase stability is maintained across cycles modulo 2π .



Figure 5. Phase Stability Testing Configuration: Multiple USRP, Shared GPSDO, Sinusoid Generator. The 4 channel receiver configuration captures the transmitted reference tone and then compares the received signals phase between the 4 channels.

Initially, there will be a random phase shift between the two channels, in accordance with the manufacturer specification [70]. Algorithms running on the system can calibrate their processing to remove this random shift. This feature is implemented through a common 10 MHz clock reference and PPS is shared between multiple chained radios, and care is taken to not reset the synthesizer's phase on transmission/reception start and stop—including when switching between transmit and receive modes. Phase stability is only available to a group of radios connected to a single edge node. Due to the behavior of the 10 MHz reference clock buffer, there are random shifts of multiples of 2π each cycle. To verify phase-coherence on transmit, a similar experiment was done in the reverse configuration. The nominal measured phase variance was 0.106 rad = 6.073°.



Figure 6. Phase difference between two channels of a single USRP device driven by a GPSDO [27].



Figure 7. Phase differences between channel pairs of a 4 channel receiver system using two USRPs (2 channels each) sharing a 10 MHz clock and PPS [27].

3.3. WISCANet-Lite

We developed a containerized deployment methodology called WISCANet-Lite [27]. WISCANet-Lite utilizes Podman [71] or Docker [72] to run multiple edge nodes on a single host computer. A control node container is also launched to provide a self-contained and self-configuring software-defined radio network (SDR-N) solution. This container solution is much lighter weight than an equivalent virtual machine, and enables higher performance applications due to reduced overhead. However, it is still primarily limited by the host system's memory and Ethernet or USB bandwidth to consume and transmit samples to the USRPs. This enables researchers to scale the system up and down as necessary for experiments. The WISCANet-Lite containers can also be deployed across multiple machines in a more typical distributed WISCANet architecture. This enables simplified distributed deployment and consistency by leveraging the benefits of containerization. There are also Ansible roles available for deploying WISCANet.

3.4. WISCANet GUI

We also developed a simple graphical user interface for WISCANet in [27]. This GUI provides access to a database of applications in an environment that can access a database of available edge nodes and their possible configurations. The interface pairs an edge node to an application and compiles a list of pairings. Once all parameters are specified for the desired nodes, these configurations are written to disk and the nodes are ready for

the Control Node to command. The interface was written using Rust and the supporting libraries, Serde, Diesel (with SQLite), and Rocket. Since this is a web application, it enables simple remote control of the system for a distributed research group.

3.5. Genie Channel

We added an optional genie channel subsystem to WISCANet in [73]. A genie channel, or perfect (near) instantaneous feedback channel, provides a mechanism to pass around information without concerns for constraints or the feasibility of a communications link. The control node provides a Network File System (NFS) directory share that is mounted on all edge nodes. This directory is accessible to the running baseband software in real time. This enables baseband software to share data through files placed in that directory. Baseband software can also wait for the existence of a file to proceed with the next cycle. This enables the researcher to simplify the design of feedback channels for experiments, as well as accurately impose constraints on the feedback, such as delay, error rate and more, while still allowing the system under test to operate over the air.

3.6. Example Lab Deployment

We deployed WISCANet on a network of Ettus X310s connected to Dell Precision Tower 3630s and Ettus B210s connected to Intel Skull Canyon NUCs [27]. This deployment is depicted in Figure 8. This particular implementation features combined X310s on each tower, providing a maximum of 4 transmit or 4 receive channels per radio. We use a GPS repeater to lock the radios in this indoor environment.

To achieve greater than 2×2 MIMO, multiple X310's must be used in tandem. Two X310s can be combined by sharing one unit's 10 MHz and PPS signals to enable a 4×4 MIMO device. This method can be extended to create larger platforms or replaced with an Octoclock-G to distribute the 10 MHz and PPS signals to all X310's in a single edge node $N \times N$ MIMO configuration. Either of these combinations can achieve the same phase-stable behavior demonstrated in the previous results.



Figure 8. This WISCANet deployment features four Ettus X310s with UBX-160s sharing 10 MHz and PPS signals disciplined by the internal GPSDO. Each edge node uses two independent X310s for higher bandwidth transmit and receive. This deployment also features four Ettus B210s as independent edge nodes. One Ettus N320 is also included for embedded processor experiments. This deployment can be used for a variety of experiments in sub-6 GHz bands, emphasizing disparate processing capability, and adaptive waveform employment.

3.7. Cross Platform Operation

WISCANet is compatible with several system architectures, including amd64, aarch32, and aarch64. We have tested WISCANet on numerous platforms, including HEDT and server-class systems, Raspberry Pis, and NVIDIA Jetsons. This enables diverse architectures for development and experimentation, and integrates unique architectural features such as the Jetson CUDA cores or integrated FPGAs on Xilinx MPSoC's.

3.8. mmWave: Millimeter Wave Frontends

WISCANet supports mmWave RFFEs to extend the Ettus USRP products beyond sub-6 frequencies to enable mmWave applications. We integrate these front-ends by using LFRX/LFTX or BasicTX/BasicRX daughterboards on X310 or N210 radios to transmit and receive directly at baseband to interface with mmWave frontends. This configuration is directly supported by UHD and does not require any extensions or modifications to the WISCANet framework or baseband software packages.

These modified USRP devices connect directly to a Sivers Evaluation Kit (EVK) or similar mmWave frontend (such as the TMYTEK BBOX). This mmWave front-end provides an up-converter to the mmWave RF frequency and an optional integrated antenna array controlled by the baseband software. The WISCANet baseband processing architecture is unchanged, and existing applications can be directly extended to mmWave operating frequencies. This provides an accessible and affordable mmWave testbed for the academic and independent research communities.

3.9. GPU Acceleration

We explore different acceleration approaches by updating the most computationallydemanding computations to run on Graphics Processing Units (GPUs). Some WISCANet applications have used general-purpose GPUs (GPGPUs) to improve processing speed using CUDA, an API developed by Nvidia to expose the GPU architecture and parallel computing elements [74]. AMD ROCm, a similar open-source software platform, can also be used in this context [75]. These tools enable characterization of novel GPU acceleration algorithms using real data.

3.10. FPGA Acceleration and Integrated Testing

WISCANet applications can also leverage the embedded FPGAs in the USRP radios via the RFNoC interface. Custom IP cores can be wrapped in a simple RFNoC block and embedded in the FPGA image. These custom blocks can then be exposed to the baseband application through WISCANet for testing and valiation of the FPGA block. This approach allows the IP cores to be validated with real RF samples and tests the block's interaction with other algorithmic components, enabling partial integration, progressive development and FPGA employment without requiring full integration of the FPGA into a real-time system.

3.11. Machine Learning

MATLAB, Python, and Julia have extensive collections of machine learning algorithm libraries. WISCANet-enabled applications can natively access these language libraries at runtime, thereby enabling rapid development and testing of machine learning algorithms using over-the-air data. Run-time and training-time challenges can be addressed using the already available GPU acceleration support, making WISCANet a suitable testbed for directly and efficiently comparing algorithm performance.

4. Over-the-Air Demonstrations

The primary objectives of WISCANet are (a) to accelerate the implementation and validation of novel radio applications and (b) to make over-the-air experimentation accessible to users that are not hardware experts. To demonstrate that WISCANet enables these capabilities, we prepared several over-the-air demonstration applications that showcase

the capabilities and ease-of-use described in the previous section. We executed these applications on the large WISCANet deployment that we maintain in the WISCA laboratory [27,30,73], which is pictured in Figure 2 and depicted in Figure 9. As a frame of reference compared to standard solutions (ASICs, FPGAs, RFSoCs, SDRs, etc.), these applications – from conception to OTA validation – were implemented in under 2 weeks each by a single graduate student. Notably, these demonstrations were all performed over-the-air on Ettus USRPs, so real-world channel effects, transmitter noise, receiver noise, RF system impairments, and front-end non-linearities are all present.



Figure 9. System architecture of the WISCANet deployment maintained in the WISCA laboratory at Arizona State University.

4.1. Distributed Mosaic Beamforming

In a previous study [73], we demonstrated a novel, distributed, multi-stage beamforming technique for distributed mesh relay networks. A simple example system that executes this technique is depicted in Figure 10. In this particular configuration, two mesh relay networks *A* and *B* execute the multi-stage beamforming technique to relay communications between a simple transmitter and receiver. Given N_a and M_b nodes in meshes *A* and *B*, this technique increases the received SNR realized at the receiver by as much as $N_a^2 M_b$ [73,76]. To quickly validate this system concept and performance bound, we implemented this system on WISCANet using Ettus X310 products. Doing so allowed us to quickly (<2 weeks) validate the system concept over-the-air before committing to a multi-million dollar research program to transition the technology to an embedded solution [73].

We implemented the application depicted in Figure 10 on WISCANet with $N_a = 4$ and $M_b = 4$ using Ettus X310s, which provide up to 4 RF channels per device. Distributed coherent applications require precisely aligned local oscillators, which is enabled intrinsicly by WISCANet's phase stability: this guarantees oscillator frequency stability if we elect to enable clock sharing.

The proposed technique is an iterative approach that requires an initial acquisition iteration. For a given transmit event: (1) the transmitter transmits to mesh A; (2) mesh A computes and applies a beamforming pre-distortion filter to the received signal; (3) mesh A re-transmits this signal to mesh B; (4) mesh B computes and applies another beamforming pre-distortion filter to the received signal; and (5) mesh B re-transmits this signal to the final receiver. Computing the pre-distortion filters requires channel state information (CSI) measurements provided by a feedback link, so in the first iteration the beamforming gain is limited. After the CSI is measured and communicated on the feedback link (in this case implemented using WISCANet's genie channel feature), the system approaches

the theoretical SNR gain of $N_a^2 M_b$. Since these demonstrations are executed over-the-air, real-world effects such as time-varying channels and noise cause some fluctuation in the actual realized performance. The SNR gain results from three independent trials of this demonstration are depicted in Figure 11.



Figure 10. Distributed mesh relay network with 4 distributed radios in each mesh *A* and *B*. These mesh relays execute a distributed, multi-stage beamforming algorithm to relay a message between the transmitter and receiver.



Figure 11. Experimental results for the 1-4-4-1 distributed mesh relay network. The y-axis represents the effective increase in SNR at the final receiver as a result of the distributed relay and multi-stage beamforming compared to the equivalent single input, single output (SISO) direct link. Channel state information is not available until after the first transmit and receive cycle, so the first measurement is limited to incoherent beamforming gain.

4.2. Ping-Pong

In addition to the WISCANet control software, we also publish several example applications in our open source distribution to help researchers develop their own custom techniques. One such application is named "ping-pong" which is a simple communications application that passes a message between an arbitrary number of radios [30]. This prebuilt network infrastructure allows users to test PHY and MAC layer techniques without having to design a network themselves, thereby reducing the barrier to entry and allowing users to focus on research rather than the tedium of networking and configuration. To demonstrate the utility of this pre-built application, we use the ping-pong application to

enable dynamic PHY and MAC layer techniques in real-time, over-the-air demonstrations. In Section 4.3, we leverage ping-pong to deploy an adaptive PHY layer technique that updates the waveform design in response to changing channel conditions. In Section 4.4, we leverage ping-pong to deploy a MAC layer technique that implements simple spectrum sensing and dynamic spectrum access.

4.3. Adaptive Waveform Development

In previous studies [27,77,78], several researchers have investigated adaptive waveform protocols that evolve in response to time-varying channel conditions. To demonstrate the accessibility of WISCANet, we tasked a graduate student with implementing these techniques around the pre-built ping-pong application.

The adaptive waveform toolbox includes a variety of basic dynamic spectrum access, frequency-hopping, dynamic forward error correction (FEC), and modulation techniques, all of which are prescribed automatically by a protocol recommendation engine (PRE). This engine optimizes the waveform properties while the system is operating, actively adapting to changes in received SNR, spectral opportunities, and channel impairments by communicating through the genie channel. The adaptive waveform toolbox leverages MATLAB's object-oriented programming capabilities to reuse the same code structure, but with dynamic algorithmic components. This toolbox is loosely depicted in Figure 12.

We designed an object-oriented Application Programming Interface (API) to interface with our large library of software-defined waveform components. In this API, components such as encoders, modulators, etc. are defined as classes that implement a standardized component interface. Applications interact with this interface instead of individual components; consequently, the components may be modified or swapped automatically by exchanging pointers, even while running an experiment.

This API enables dynamic component selection in response to user inputs or changes in the environment [77,78]. While the ping-pong system is operating, we inject additional noise to degrade the SNR at both radios. This corrupts the decoded message, thereby limiting channel capacity, which the PRE assesses and adapts to by decreasing the coding rate and changing the modulation for subsequent transmit and receive cycles. Through the adaptation, the subsequent communications are decoded without error even with the injected noise. The same adaptation capability functions to switch between single-carrier and OFDM modulations, or OFDM parameters to adapt to changing channel delay spreads. These results are depicted in Figure 13.



Figure 12. Block diagram of the ping-pong application with integrated waveform development toolbox. Two radios alternate transmitting and receiving a message while simultaneously performing channel measurements and spectral sensing. A protocol recommendation engine automatically optimizes the waveform and frequency allocation to adapt to changes in received SNR, spectral opportunities, and channel impairments. WISCANet allows users to test these kinds of PHY and MAC layer techniques without having to design a network themselves, thereby reducing the barrier to entry and allowing users to focus on research rather than the tedium of networking and configuration.



Figure 13. Console output for the ping-pong communications system when noise is artificially injected into the system. Observe the degrading SNR, causing significant errors in the received signal. The PRE then automatically adjusts the communications protocol and leverages the Object Oriented API to swap out software-defined components and change parameters for the subsequent transmission, which is then decoded without error. This demonstration, like the others, is performed over-the-air in our indoor laboratory environment, so this demonstration includes a real channel and real noise sources.

4.4. Dynamic Spectrum Access

In the previous section, we demonstrated that WISCANet and the ping-pong application enable rapid validation of dynamic PHY layer techniques. In this section, we further demonstrate that the system also supports validation of dynamic MAC layer techniques by implementing a simple dynamic spectrum access protocol. This implementation is an extension to the same results cited previously [27].

We deployed several algorithms to automatically detect vacancies in time and frequency and find appropriate time-frequency slots for opportunistic spectral access. B5G and 6G systems demand better techniques for opportunistic time-frequency resource usage. 5G systems already do this by employing Bandwidth Parts and other co-operative access algorithms. The following results demonstrate the utility of WISCANet in quickly developing, implementing, and validating novel MAC layer techniques for B5G and 6G applications.

In the first over-the-air demonstration [27], we configure 3 spectrum users to periodically transmit at different, fixed carrier frequencies. A fourth user listens to these transmissions for several frames and determines an appropriate center frequency and bandwidth that does not interfere with the other 3 transmitters. This user then executes the "ping-pong" communications system in the discovered open spectral vacancy.

In the second over-the-air demonstration [27], we configure the same 3 spectrum users to periodically transmit at different, fixed carrier frequencies in adjacent frequency bins, such that these 3 users span the majority of the spectrum and do not leave sufficient spectral vacancy for frequency-division-only use. The fourth user listens to the environment for several cycles and selects an available time-frequency slot in which to execute the "pingpong" communications system without interfering with the other users. The chosen window is at the same carrier frequency as one of the other users but does not overlap with the incumbent transmissions. The system architecture and over-the-air results are presented in Figure 14.



Figure 14. Block diagram of the dynamic spectral access configuration. Three users labeled "others" operate in different spectral allocations with arbitrary timing. The two users engaged in the pingpong application constantly measure the spectrum, identify opportunities, and use these openings to communicate. The goal of this demonstration is to highlight how WISCANet and the pre-built ping-pong application enable rapid implementation and validation of novel PHY and MAC layer techniques over-the-air.

4.5. 60 GHz Communications System

To support higher frequency ranges, WISCANet deployments may be augmented by either mixing directly from baseband or superheterodyning inside the equipped RFFEs. In our specific deployment (pictured in Figure 2), we have executed over-the-air applications as high as 71 GHz. To support 6G research, we plan to extend support into the THz band; unfortunately, commercial THz SDRs are not widely available, so we implemented a superheterodyning approach similar to the technique discussed in [31].

Using the Sivers EVK in combination with USRP N210's with LFRX and LFTX daughterboards, we implemented a simple mmWave communications system leveraging the adaptive waveform development kit. Operating at 60 GHz adds several challenges that are often be ignored at lower frequencies; for lower sampling rates, the carrier frequency offset (CFO) can become an appreciable fraction of the bandwidth, thus applications require some form of robust CFO correction. To address this issue, we added a robust CFO estimation and correction technique based on [79]. We add an underlying tone to the transmitted waveform as a fiducial for the CFO estimator. After the CFO correction is complete, a simple DC-blocking IIR filter is constructed and applied to remove the tone. With these two additions, the WDK performs comparably to the sub-6 GHz implementations at the new 60.48 GHz carrier frequency. These challenges were easy to identify and resolve thanks to the flexible debugging nature of WISCANet, and easy to address because of the ease of adapting code that is running on-the-air.

We adapted the "ping-pong" communications demonstration and adaptive waveform toolbox to demonstrate flexible waveform implementation on NR Frequency Range 2 (FR2) systems. This showcases WISCANet's frequency flexibility and support for increasing frequency ranges. In Figures 15–19, we plot a snapshot of the power spectral densities and received symbols. The carrier frequency for the experiment is the WiGig Channel 2 at 60.48 GHz. We used a 10 MHz sampling rate and oversampled the BPSK waveform by 5 to achieve an occupied bandwidth of about 2 MHz. This allows plenty of flexibility in correcting CFO and reduced the risk of aliasing. Between Figures 17 and 18, we observe some residual carrier frequency offset in the symbols, which reduces the effectiveness of the applied MMSE equalizer. A more accurate and computationally-expensive fine frequency estimation algorithm could help reduce these residual effects.



Figure 15. Snapshot of the over-the-air, received power spectral density in the WiGig Channel 2 band at 60.48 GHz.



Figure 16. Snapshot of the over-the-air, received power spectral density in the WiGig Channel 2 band at 60.48 GHz after matched filtering and CFO correction.



Figure 17. Received training symbols after CFO correction.



Figure 18. Received message symbols after CFO correction.



Figure 19. Received message symbols after CFO correction and equalization.

Multi-function systems are an emerging class of high-performance, highly cooperative techniques that promise many of the performance enhancements demanded by B5G and 6G applications. Implementation and validation of these complex techniques was one of the primary motivations in developing WISCANet. In the following demonstrations, we highlight how WISCANet enables us to quickly advance research in this field while traditional solutions are otherwise prohibitively expensive and difficult.

We deployed two over-the-air implementations of a system that enables joint radar, communications, positioning, navigation, and timing (JRCPNT) [6,8]. The system simultaneously executes 5G communications, mono- and bi-static radar, and time-of-arrival (ToA) estimation to enable precise ranging and time synchronization. We used the same laboratory deployment depicted in Figure 9 with two edge nodes using Ettus X310s.

The first configuration operates with 10 MHz bandwidth at 915 MHz center frequency in the ISM band using standard dipole antennas. This configuration was used to test the radar algorithm functionality. A target emulator "traveling" with a ten meter per second velocity at a range of 10 km was emulated over-the-air in a high-SNR, low-clutter environment. The results depicted in Figure 20. To identify potential targets, we apply a coarse matched filter in the time domain to identify the peaks associated with the direct path (blue) and the return path (red). An FFT-based time-frequency matched filter is then applied around the return peak to identify the relative velocity and range of the target. In the given environment we can clearly identify the target and correctly estimate the parameters.



Figure 20. Initial over-the-air results under the first experimental configuration.

The second configuration operates with 100 MHz bandwidth at 5.8 GHz center frequency. Horn antennas were used because they were the most readily available option for the given frequency range, although directional antennas are not required to deploy this system [80]. This configuration was used to evaluate the functionality of the ToA, ranging, and synchronization features. In Figure 21, The ToA estimation standard deviation is plotted against the CRLB and fundamental resolution. After estimating the ToA, the system executes a PNT ranging and synchronization algorithm described in [13]. The results are plotted in Figures 22 and 23 as a function of integrated SNR. With this experimental configuration, we approach centimeter-level ranging precision and nanosecond-level time synchronization given sufficient signal power and waveform integration.

While this implementation significantly underperforms custom implementations of comparable systems [13], we were able to generate a proof-of-concept prototype that validates the functionality of the various sub-systems in under a week. For reference, the final results presented in [13] are the culmination of 4 graduate students, 18 months of effort, and \$1M of funding. While WISCANet does not promise the performance of a fully-custom solution, it does boast incredible speed, accessibility, and efficiency in developing functional prototypes for a variety of RF applications.



Figure 21. Over-the-air time-of-arrival estimation standard deviation, plotted against the CRLB and fundamental resolution.



Figure 22. Over-the-air ranging results, demonstrating sub-10-cm ranging precision.



Figure 23. Over-the-air synchronization results, demonstrating sub-ns synchronization.

5. Conclusions

In this work, we describe the architecture and capabilities of the WISCANet softwaredefined radio network (SDR-N) control software. We demonstrate that this technology offers an accessible and affordable rapid-development platform for over-the-air (OTA) sub-6 and mmWave experiments. We present several example applications to demonstrate the features and capabilities of WISCANet, including several integrated sensing and communications (ISAC) applications. To further emphasize the accessibility of this technology, we release the WISCANet source code and the supporting research packages under the Lesser GNU Public License Version 3.0 (LGPLv3.0). This software may be found at https://github.com/WISCA/ (accessed on 30 September 2022).

Author Contributions: Conceptualization, J.H., A.H. and D.W.B.; Data curation, J.H.; Formal analysis, J.H., I.L. and O.M.; Funding acquisition, A.H. and D.W.B.; Investigation, J.H., I.L. and O.M.; Methodology, J.H., A.H. and D.W.B.; Project administration, A.H. and D.W.B.; Resources, A.H. and D.W.B.; Software, J.H., I.L., O.M. and H.Y.; Supervision, A.H. and D.W.B.; Validation, J.H. and I.L.; Visualization, J.H. and A.H.; Writing—original draft, J.H., A.H. and I.L.; Writing—review & editing, J.H., A.H., I.L., O.M., H.Y. and D.W.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Software is available at https://github.com/wisca/ (accessed on 30 September 2022).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

5G	fifth generation
ABI	Application Binary Interface
API	Application Programming Interface
B5G	beyond-5G
COTS	common off the shelf
CSI	channel state information
CUDA	Compute Unified Device Architecture
EVK	Evaluation Kit
FPGA	Field Programmable Gate Array
FR1	NR Frequency Range 1
FR2	NR Frequency Range 2
GPS	Global Positioning System
GPSDO	GPS Disciplined Oscillator
GUI	Graphical User Interface
IP	Intellectual Property
JRCPNT	joint radar, communications, positioning, navigation, and timing
LGPLv3.0	Lesser GNU Public License Version 3.0
MAC	Media Access Control Layer
MIMO	multiple-input, multiple-output
NFS	Network File System
NI	National Instruments
NR	New Radio
OTA	over-the-air
PHY	Physical Layer
RENEW	Reconfigurable Ecosystem for Next-generation End-to-end Wireless

RF	radio-frequency
RFFE	RF front-end
RFSoC	RF system-on-chip
SISO	single input, single output
SDR	software-defined radio
SDR-N	software-defined radio network
SFFPC	Small Form Factor PC
SSH	Secure Shell
ТСР	Transmission Control Protocol
UDP	User Datagram Protocol
UHD	USRP Hardware Driver
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
WISCA	The Center for Wireless Information Systems and Computational Architectures
WISCANet	WISCA Software-Defined Radio Network
YAML	YAML Ai not Markup Language

References

- FG-NET-2030. Network 2030 a Blueprint of Technology, Applications and Market Drivers towards the Year 2030 and Beyond. Available online: https://www.itu.int/en/ITU-T/focusgroups/net2030/Documents/White_Paper.pdf (accessed on 30 September 2022).
- 2. Döhler, A. Report from the next generation mobile networks alliance. *IEEE Netw.* 2021, 35, 3. [CrossRef]
- Jiang, W.; Han, B.; Habibi, M.A.; Schotten, H.D. The road towards 6G: A comprehensive survey. *IEEE Open J. Commun. Soc.* 2021, 2, 334–366. [CrossRef]
- 4. Cui, Y.; Liu, F.; Jing, X.; Mu, J. Integrating sensing and communications for ubiquitous IoT: Applications, trends, and challenges. *IEEE Netw.* **2021**, *35*, 158–167. [CrossRef]
- 5. Paul, B.; Chiriyath, A.R.; Bliss, D.W. Survey of RF communications and sensing convergence research. *IEEE Access* 2016, 5, 252–270. [CrossRef]
- Herschfelt, A.; Chiriyath, A.R.; Srinivas, S.; Bliss, D.W. An introduction to spectral convergence: Challenges and paths to solutions. In Proceedings of the 2021 1st IEEE International Online Symposium on Joint Communications & Sensing (JC&S), Dresden, Germany, 23–24 February 2021; pp. 1–6.
- Chiriyath, A.R.; Paul, B.; Bliss, D.W. Radar-communications convergence: Coexistence, cooperation, and co-design. *IEEE Trans. Cogn. Commun. Netw.* 2017, 3, 1–12. [CrossRef]
- 8. Herschfelt, A. Vehicular RF Convergence: Simultaneous Radar, Communications, and PNT for Urban Air Mobility and Automotive Applications. In Proceedings of the 2012 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020.
- 9. Chiriyath, A.R. Fundamental Limits on Performance for Cooperative Radar-Communications Coexistence. Ph.D. Thesis, Arizona State University, Tempe, AZ, USA, 2018.
- Uhrie, R.; Bliss, D.W.; Chakrabarti, C.; Ogras, U.Y.; Brunhaver, J. Machine understanding of domain computation for domainspecific system-on-chips (dssoc). In *Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation* 2019; SPIE: Bellingham, WA, USA, 2019; Volume 11015, pp. 180–187.
- Chiriyath, A.R.; Herschfelt, A.; Srinivas, S.; Bliss, D.W. Technological Advances to Facilitate Spectral Convergence. In Proceedings of the 2021 55th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 31 October–3 November 2021; pp. 623–628.
- 12. Holt, C.M. Novel Learning-Based Task Schedulers for Domain-Specific SoCs. Ph.D. Thesis, Arizona State University, Tempe, AZ, USA, 2020.
- 13. Herschfelt, A. Simultaneous Positioning and Communications: Hybrid Radio Architecture, Estimation Techniques, and Experimental Validation. Ph.D. Thesis, Arizona State University, Tempe, AZ, USA, 2019.
- 14. Gutierrez, R.M.; Yu, H.; Chiriyath, A.R.; Gubash, G.; Herschfelt, A.; Bliss, D.W. Joint Sensing and Communications Multiple-Access System Design and Experimental Characterization. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–8.
- Mahmood, N.H.; Alves, H.; López, O.A.; Shehab, M.; Osorio, D.P.M.; Latva-Aho, M. Six key features of machine type communication in 6G. In Proceedings of the 2020 2nd 6G Wireless Summit (6G SUMMIT), Levi, Finland, 17–20 March 2020; pp. 1–5.
- 16. Rodrigues, T.K.; Liu, J.; Kato, N. Offloading decision for mobile multi-access edge computing in a multi-tiered 6G network. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 1414–1427. [CrossRef]
- 17. Morocho-Cayamcela, M.E.; Lee, H.; Lim, W. Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions. *IEEE Access* 2019, 7, 137184–137206. [CrossRef]

- 18. Nawaz, S.J.; Sharma, S.K.; Wyne, S.; Patwary, M.N.; Asaduzzaman, M. Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future. *IEEE Access* 2019, *7*, 46317–46350. [CrossRef]
- Chowdhury, M.Z.; Shahjalal, M.; Ahmed, S.; Jang, Y.M. 6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions. *IEEE Open J. Commun. Soc.* 2020, 1, 957–975. [CrossRef]
- Shafin, R.; Liu, L.; Chandrasekhar, V.; Chen, H.; Reed, J.; Zhang, J.C. Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G. *IEEE Wirel. Commun.* 2020, 27, 212–217. [CrossRef]
- Strinati, E.C.; Barbarossa, S.; Gonzalez-Jimenez, J.L.; Ktenas, D.; Cassiau, N.; Maret, L.; Dehos, C. 6G: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication. *IEEE Veh. Technol. Mag.* 2019, 14, 42–50. [CrossRef]
- Pan, C.; Ren, H.; Wang, K.; Kolb, J.F.; Elkashlan, M.; Chen, M.; Di Renzo, M.; Hao, Y.; Wang, J.; Swindlehurst, A.L.; et al. Reconfigurable intelligent surfaces for 6G systems: Principles, applications, and research directions. *IEEE Commun. Mag.* 2021, 59, 14–20. [CrossRef]
- 23. Basar, E. Reconfigurable intelligent surface-based index modulation: A new beyond MIMO paradigm for 6G. *IEEE Trans. Commun.* **2020**, *68*, 3187–3196. [CrossRef]
- Alexandropoulos, G.C.; Lerosey, G.; Debbah, M.; Fink, M. Reconfigurable intelligent surfaces and metamaterials: The potential of wave propagation control for 6G wireless communications. *arXiv* 2020, arXiv:2006.11136.
- Akeela, R.; Dezfouli, B. Software-defined Radios: Architecture, state-of-the-art, and challenges. Comput. Commun. 2018, 128, 106–125. [CrossRef]
- 26. Zhang, J.; Jia, J.; Zhang, Q.; Lo, E.M. Implementation and evaluation of cooperative communication schemes in software-defined radio testbed. In Proceedings of the 2010 Proceedings IEEE INFOCOM, San Diego, CA, USA, 14–19 March 2010; pp. 1–9.
- Holtom, J.; Gubash, G.; Herschfelt, A.; Ma, O.; Standage-Beier, W.; Yu, H.; Bliss, D.W. Rapid implementation and demonstration of radio applications using wiscanet. In Proceedings of the 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 13–16 September 2021; pp. 1500–1505.
- 28. Xilinx Adaptive SoCs. Available online: https://www.xilinx.com/products/silicon-devices/soc.html (accessed on 30 September 2022).
- Amakawa, S.; Aslam, Z.; Buckwater, J.; Caputo, S.; Chaoub, A.; Chen, Y.; Corre, Y.; Fujishima, M.; Ganghua, Y.; Gao, S.; et al. White Paper on RF Enabling 6G—Opportunities and Challenges from Technology to Spectrum. 2021. Available online: https://biblio.ugent.be/publication/8704523 (accessed on 30 September 2022).
- Yu, H.; Lee, H.; Gutierrez, R.M.; Herschfelt, A.; Bliss, D.W. WISCA SDR Network. In Proceedings of the MILCOM 2017–2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25 October 2017; pp. 750–755.
- Abdellatif, H.; Ariyarathna, V.; Petrushkevich, S.; Madanayake, A.; Jornet, J.M. A Real-Time Ultra-broadband Software-Defined Radio Platform for Terahertz Communications. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, 2–5 May 2022; pp. 1–2. [CrossRef]
- Schifferer, B.; Titericz, G.; Deotte, C.; Henkel, C.; Onodera, K.; Liu, J.; Tunguz, B.; Oldridge, E.; De Souza Pereira Moreira, G.; Erdem, A. GPU accelerated feature engineering and training for recommender systems. In *Proceedings of the Recommender Systems Challenge* 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 16–23.
- Mangiatordi, F.; Pallotti, E. A GPU accelerated framework for monitoring LTE/5G interference to DVB-T systems. In Proceedings of the 2021 AEIT International Annual Conference (AEIT), Milan, Italy, 4–8 October 2021; pp. 1–6.
- Talib, M.A.; Majzoub, S.; Nasir, Q.; Jamal, D. A systematic literature review on hardware implementation of artificial intelligence algorithms. J. Supercomput. 2021, 77, 1897–1938. [CrossRef]
- 35. Steinkraus, D.; Buck, I.; Simard, P. Using GPUs for machine learning algorithms. In Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea, 31 August–1 September 2005; pp. 1115–1120.
- 36. Fernandes, S.; Karmouch, A. Vertical mobility management architectures in wireless networks: A comprehensive survey and future directions. *IEEE Commun. Surv. Tutor.* **2010**, *14*, 45–63. [CrossRef]
- 37. Dinh, C.T.; Tran, N.H.; Nguyen, M.N.; Hong, C.S.; Bao, W.; Zomaya, A.Y.; Gramoli, V. Federated learning over wireless networks: Convergence analysis and resource allocation. *IEEE/ACM Trans. Netw.* **2020**, *29*, 398–409. [CrossRef]
- Qi, Q.; Chen, X.; Zhong, C.; Zhang, Z. Integration of energy, computation and communication in 6G cellular internet of things. IEEE Commun. Lett. 2020, 24, 1333–1337. [CrossRef]
- Khan, L.U.; Yaqoob, I.; Imran, M.; Han, Z.; Hong, C.S. 6G wireless systems: A vision, architectural elements, and future directions. IEEE Access 2020, 8, 147029–147044. [CrossRef]
- 40. Li, X.; Wang, Q.; Liu, M.; Li, J.; Peng, H.; Piran, M.J.; Li, L. Cooperative wireless-powered NOMA relaying for B5G IoT networks with hardware impairments and channel estimation errors. *IEEE Internet Things J.* **2020**, *8*, 5453–5467. [CrossRef]
- Le, H.A.; Van Chien, T.; Nguyen, T.H.; Choo, H.; Nguyen, V.D. Machine Learning-Based 5G-and-Beyond Channel Estimation for MIMO-OFDM Communication Systems. *Sensors* 2021, 21, 4861. [CrossRef] [PubMed]
- 42. Yang, P.; Xiao, Y.; Xiao, M.; Li, S. 6G wireless communications: Vision and potential techniques. *IEEE Netw.* 2019, 33, 70–75. [CrossRef]
- 43. Shah, S.D.A.; Gregory, M.A.; Li, S. Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. *IEEE Access* 2021, *9*, 10903–10924. [CrossRef]
- 44. Haneche, H.; Ouahabi, A.; Boudraa, B. New mobile communication system design for Rayleigh environments based on compressed sensing-source coding. *IET Commun.* **2019**, *13*, 2375–2385. [CrossRef]

- 45. Haneche, H.; Ouahabi, A.; Boudraa, B. Compressed sensing-speech coding scheme for mobile communications. *Circuits Syst. Signal Process.* **2021**, *40*, 5106–5126. [CrossRef]
- Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Niyato, D.; Dobre, O.; Poor, H.V. 6G Internet of Things: A comprehensive survey. *IEEE Internet Things J.* 2021, *9*, 359–383. [CrossRef]
- Dao, N.N.; Pham, Q.V.; Tu, N.H.; Thanh, T.T.; Bao, V.N.Q.; Lakew, D.S.; Cho, S. Survey on aerial radio access networks: Toward a comprehensive 6G access infrastructure. *IEEE Commun. Surv. Tutor.* 2021, 23, 1193–1225. [CrossRef]
- 48. Ettus Research—The Leader in Software Defined Radio (SDR). Available online: https://www.ettus.com/ (accessed on 30 September 2022).
- Instruments, N. Software Defined Radios. Available online: https://novatorsolutions.com/products/hugin4000/?gclid= EAIaIQobChMIma_H8L6l-gIV16uWCh22OQD_EAAYASAAEgIRdvD_BwE (accessed on 30 September 2022).
- 50. USRP Hardware Driver (UHD)—Ettus Research. Available online: https://www.ettus.com/sdr-software/uhd-usrp-hardwaredriver/ (accessed on 30 September 2022).
- MathWorks. USRP Support from Communications Toolbox. Available online: https://jp.mathworks.com/hardware-support/ usrp.html (accessed on 30 September 2022).
- 52. GNURadio: The Free & Open Software Radio Ecosystem. 2020. Available online: https://www.gnuradio.org/ (accessed on 30 September 2022).
- 53. GNURadio. 2020. Available online: https://github.com/gnuradio/gnuradio (accessed on 30 September 2022).
- 54. Vilches, T.; Dujovne, D. GNUradio and 802.11: Performance evaluation and limitations. *IEEE Netw.* 2014, 28, 27–31. [CrossRef]
- 55. Pothosware. Welcome to the SoapySDR Project. 2020. Available online: https://www.pothosware.com/ (accessed on 30 September 2022).
- 56. Pothosware. Pothosware, 2020. Available online: https://github.com/pothosware (accessed on 30 September 2022).
- Shepard, C.; Yu, H.; Anand, N.; Li, E.; Marzetta, T.; Yang, R.; Zhong, L. Argos: Practical Many-Antenna Base Stations. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 53–64.
- Doost-Mohammady, R.; Bejarano, O.; Zhong, L.; Cavallaro, J.R.; Knightly, E.; Mao, Z.M.; Li, W.W.; Chen, X.; Sabharwal, A. RENEW: Programmable and Observable Massive MIMO Networks. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 28–31 October 2018; pp. 1654–1658.
- 59. University, R. RENEW: Reconfigurable Eco-system for Next-generation End-to-end Wireless. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 28–31 October 2020.
- Anand, N.; Aryafar, E.; Knightly, E.W. WARPlab: A Flexible Framework for Rapid Physical Layer Design. In Proceedings of the 2010 ACM Workshop on Wireless of the Students, by the Students, for the Students, Chicago, IL, USA, 20–24 September 2010; pp. 53–56.
- 61. Duarte, M.; Murphy, P.; Hunter, C.; Gupta, S.; Sabharwal, A. WARPLab: Multi-Node Prototyping with Real Wireless Data. Available online: https://warpproject.org/trac/raw-attachment/wiki/2010WARPLab (accessed on 30 September 2022).
- Minden, G.J.; Evans, J.B.; Searl, L.; DePardo, D.; Petty, V.R.; Rajbanshi, R.; Newman, T.; Chen, Q.; Weidling, F.; Guffey, J.; et al. KUAR: A Flexible Software-Defined Radio Development Platform. In Proceedings of the 2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, Ireland, 17–20 April 2007; pp. 428–439. [CrossRef]
- Samoohi, S.; Graff, C. Challenges of Using Software Defined Radio Platforms in Validating Cognitive Network Concepts. In Proceedings of the 2010 IEEE International Conference on Wireless Information Technology and Systems, Honolulu, HI, USA, 28 August–3 September 2010; pp. 1–4.
- Asadi, A.; Mancuso, V.; Gupta, R. An SDR-based Experimental Study of Outband D2D Communications. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [CrossRef]
- Yau, S.; Ge, L.; Hsieh, P.C.; Hou, I.H.; Cui, S.; Kumar, P.; Ekbal, A.; Kundargi, N. WiMAC: Rapid Implementation Platform for User Definable MAC Protocols Through Separation. SIGCOMM Comput. Commun. Rev. 2015, 45, 109–110. [CrossRef]
- 66. Shearman, S.; Kimery, J. Software Defined Radio Prototyping Platforms Enable a Flexible Approach to Design [Application Notes]. *IEEE Microw. Mag.* 2012, 13, 76–80. [CrossRef]
- 67. Leferman, M.J. Rapid Prototyping Interface for Software Defined Radio Experimentation. Ph.D. Thesis, Worcester Polytechnic Institute, Worcester, MA, USA, 2010.
- 68. Polese, M.; Restuccia, F.; Gosain, A.; Jornet, J.; Bhardwaj, S.; Ariyarathna, V.; Mandal, S.; Zheng, K.; Dhananjay, A.; Mezzavilla, M.; et al. MillimeTera: Toward a large-scale open-source mmWave and terahertz experimental testbed. In Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems, Los Cabos, Mexico, 25 October 2019; pp. 27–32.
- 69. Yanik, M.E.; Wang, D.; Torlak, M. Development and demonstration of MIMO-SAR mmWave imaging testbeds. *IEEE Access* 2020, *8*, 126019–126038. [CrossRef]
- Synchronization and MIMO Capability with USRP Devices. Available online: https://kb.ettus.com/Synchronization_and_ MIMO_Capability_with_USRP_Devices (accessed on 20 January 2021).
- 71. Podman. Red Hat. Available online: https://podman.io/ (accessed on 30 September 2022).
- 72. Docker Inc.. Empowering App Development for Developers. Available online: https://www.docker.com/ (accessed on 30 September 2022).

- Holtom, J.; Ma, O.; Herschfelt, A.; Landon, D.G.; Bliss, D.W. Distributed Beamforming Techniques for Flexible Communications Networks. In Proceedings of the 2021 55th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 28 April 2021; pp. 400–404. [CrossRef]
- 74. CUDA Toolkit—Free Tools and Training. 2022. Available online: https://developer.nvidia.com/cuda-toolkit (accessed on 30 September 2022).
- 75. New AMD ROCM Information Portal—ROCM V4.5 and above. Available online: https://rocmdocs.amd.com/en/latest/ Current_Release_Notes/Current-Release-Notes.html (accessed on 30 September 2022).
- Bliss, D.W.; Kraut, S.; Agaskar, A. Transmit and receive space-time-frequency adaptive processing for cooperative distributed MIMO communications. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 5221–5224.
- 77. Bhat, G.; Srinivas, S.; Chagari, V.; Park, J.; McGiffen, T.; Lee, H.; Bliss, D.W.; Chakrabarti, C.; Ogras, U.Y. Fluid Wireless Protocols: Energy-Efficient Design and Implementation. In Proceedings of the 15th IEEE/ACM Symposium on Embedded Systems for Real-Time Multimedia, Seoul, Korea, 15–20 October 2017; pp. 22–31. [CrossRef]
- McGiffen, T.; Srinivas, S.; Bliss, D.W. A protocol development kit for wireless systems. In Proceedings of the 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, 5–8 February 2018; pp. 389–394. [CrossRef]
- 79. Candan, C. A Method For Fine Resolution Frequency Estimation From Three DFT Samples. *IEEE Signal Process. Lett.* 2011, 18, 351–354. [CrossRef]
- Herschfelt, A.; Yu, H.; Wu, S.; Srinivas, S.; Li, Y.; Sciammetta, N.; Smith, L.; Rueger, K.; Lee, H.; Chakrabarti, C.; et al. Joint positioning-communications system design and experimental demonstration. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019; pp. 1–6.