SurfOS: Towards an Operating System for Programmable Radio Environments

Ruichun Ma
ruichun.ma@yale.edu

Lili Qiu
liliqiu@microsoft.com

Wenjun Hu
wenjun.hu@cantab.net

Abstract

Programmable radio environments with metasurfaces introduce signal-level programmability to wireless networks, providing various services such as connectivity enhancement, coverage extension, sensing, security protection, and wireless powering. Next-generation wireless networks are set to widely deploy metasurfaces. However, the current *one-system-per-use-case* approach cannot scale with wide-ranging hardware designs and surface-aided applications. This paper presents a vision, SurfOS, a metasurface operating system for programmable radio environments. SurfOS aims to orchestrate heterogeneous surface hardware and provide diverse services for user-level applications. We discuss the challenges of building such a system, potential abstraction layers, and open research problems. Our early-stage implementation demonstrates the feasibility and benefits of this approach.

CCS Concepts

• **Networks** → **Wireless local area networks**; **Programmable networks**; **Network management**.

Keywords

Wireless Networks, Operating System, Metasurfaces

ACM Reference Format:
Ruichun Ma, Lili Qiu, and Wenjun Hu. 2024. SurfOS: Towards an Operating System for Programmable Radio Environments. In *The 23rd ACM Workshop on Hot Topics in Networks (HotNets '24), November 18–19, 2024, Irvine, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3696348.3696861

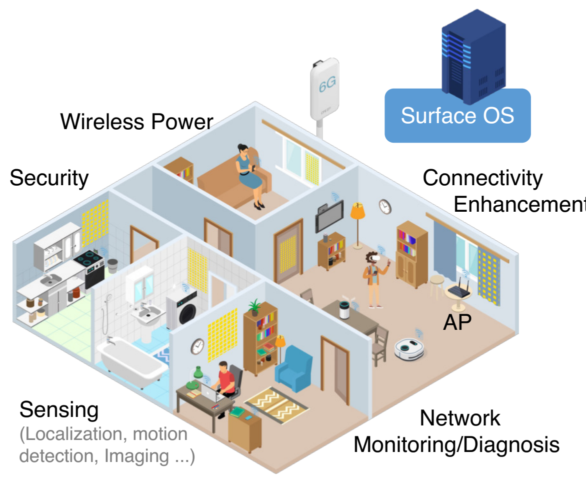


Figure 1: An illustrative deployment scenario. SurfOS manages diverse surface hardware and services intelligently.

1 Introduction

The pursuit for network programmability has a long history and great successes. It has led to Software Defined Networking (SDN) [30, 50, 71], revamping network design and management first for enterprise and data center networks [19, 38], then to wide area networks [34, 36] and radio access networks (RAN) for cellular and Wi-Fi [16, 33, 59, 60]. These efforts have reduced network management complexity while improving reliability. Managing wireless signal propagation is the missing piece over the last hop, yet with major impacts on the end-to-end service quality and user experience.

To fill in the missing piece, the visions for programmable radio environments seek to provide signal-level programmability [17, 44, 58, 67, 69], with metasurfaces serving as general-purpose hardware enablers. Actuated or configured by software, surfaces offer a variety of services, including connectivity enhancement, coverage extension, sensing, security protection, wireless powering and more. Compared with conventional networking infrastructure, surfaces operating at the signal level offer unique benefits of being standard-agnostic [47, 73], low-power or zero-power [37, 48, 57], low-complexity and low-cost [46, 48]. Therefore, surfaces are being considered as 6G infrastructure. (section 2)

From a network programmability perspective, surfaces introduce novel primitives to program wireless signal propagation. However, unlike programmable switches and middleboxes that manage packet flows without changing the physical connectivity, surfaces fundamentally alter electromagnetic wave propagation in the environments, which boosts connectivity and various other wireless signal-based applications. Therefore, existing SDN management frameworks

HotNets '24, November 18–19, 2024, Irvine, CA, USA

ACM ISBN 979-8-4007-1272-2/24/11
https://doi.org/10.1145/3696348.3696861

can not cover surfaces. Instead, we need a *propagation environment manager* to coordinate all wireless services. Existing surface prototype systems each proposes customized hardware and software control modules for specific wireless problems. Aside from redundant development efforts, this leads to several issues. First, this results in heterogeneous hardware managed by non-standard software. The inconsistent software-hardware interfaces hinder the interoperation of surface hardware and the development of a universal control plane. Second, it impedes the seamless integration of surface-aided applications, i.e., efficiently providing multiple services (such as joint communication and sensing for 6G) without interfering with one another. Lastly, there is still a gap between enhancing signal-level metrics and improving application-level performance perceived by end users. Thus, the current *one-system-per-use-case* approach does not scale with the diversity in surface hardware, surface-supported services, or user-level applications. (subsection 2.1)

Our Vision. This paper presents SurfOS, a metasurface operating system for programmable radio environments. We use the term "operating system" figuratively, as in NOX (network operating system) [32] for SDN and HomeOS [28] for IoT applications, instead of literally for a traditional OS like Linux. In contrast to the *one-system-per-use-case* approach, we propose a general-purpose system to manage heterogeneous surface hardware within one or across multiple environments. Operating at the edge or in the cloud, SurfOS multiplexes diverse services over shared hardware for a range of applications (Figure 1). We decouple the data plane (signal alteration) from the control plane (surface control) with unified APIs, akin to SDN, allowing one universal control plane that provides high-level service abstractions. Network or building administrators ("app developers/users" of SurfOS) can develop surface applications to meet end users' demands. The nature of metasurfaces and wireless networks incurs unique challenges, originating from the shared radio environment, surfaces' signal-level functionality, and their role as last-hop infrastructure serving end user applications. (section 3)

SurfOS tackles these challenges with three abstraction layers tailored to surface hardware and operations: (i) Hardware manager, drivers providing unified programming interfaces for heterogeneous surface hardware. (ii) Surface orchestrator, a universal central control plane providing high-level APIs of service abstractions. (iii) Service broker, a daemon that transparently serves existing wireless applications and coexists with new surface-native applications. The proposed abstractions also lay out clear APIs that lend to workflow automation with Large Language Models (LLMs) (subsection 3.4). We can extend recent proposals of LLMs-assisted network management [49, 51, 62] to wireless networks. This helps reduce manual efforts to develop applications or (new) surface hardware drivers and, thus, facilitates system adoption. Simulation results with our early-stage implementation show the feasibility and benefits of SurfOS: (1) enabling explicit and flexible system trade-offs among cost, size, re-configurability, deployment complexity; (2) supporting communication and sensing simultaneously with *a single* shared surface configuration (an array of phase shift values); (3) translating user demands to service API calls (section 4).

SurfOS provides OS-like functionality for both surface hardware management and application development, while facilitating workflow automation. To a minimal extent, it is a reusable control plane across surfaces for individual use cases; It should effortlessly scale to multiple services atop one or multiple nearby surfaces, or even across sites. SurfOS can be a service from ISPs (Internet service providers), a module of Cloud RAN, or a standalone system from a new service provider. These possibilities map to different business models and network architectures.

2 The case for SurfOS

The visions of *programmable radio environments* [43, 44, 67] challenge the typical assumption that wireless channel responses are uncontrollable and fixed constraints to link- and network-level system designs. Another thread of work [17, 27, 58], termed *reconfigurable intelligent surface* (RIS), shares the idea and emphasizes using surfaces as hardware.

Substantial industry interest. Since several white papers argued for including surfaces in 6G (e.g., [5, 6, 65]), there has been substantial industry interest, including standardization efforts [9, 10], industry testbeds for surfaces (e.g., [7, 8, 53]), and early commercial surface hardware [2, 4]. These developments suggest that surfaces will be widely deployed in next-generation wireless networks. Compared with alternatives for coverage provisioning, such as more APs, repeaters, mesh routers, surfaces address the root causes of wireless problems by controlling signal propagation behavior. This offers several advantages: (1) solving common issues across standards and networks as shared infrastructure (e.g., [47, 73]), (2) low-power operations via reverse-biased diodes or zero-power via fully passive metallic patterns (e.g., [37, 48, 57]), (3) low system complexity and low cost (e.g., [46, 48]).

Existing surface prototypes. Metasurfaces (or smart surfaces) are two-dimensional artificial material structures, often constructed as an array of sub-wavelength *elements* (also called *meta-atoms* or *units*). Each surface element is composed of circuit components and metallic patterns, to effectively capture and actuate passing electromagnetic waves [26, 39, 56]. Existing end-to-end systems have explored various signal control modalities: phase, amplitude, polarization, frequency [15, 21, 29, 43, 47], and impedance matching [46].

Surface Systems	Freq Band	Signal Control	Mode	Re-configurable	Cost ($)
LAIA [43]	2.4 GHz	Phase	T	✓	/
RFocus [15]	2.4 GHz	Amplitude	T & R	✓	/
LLAMA [21]	2.4 GHz	Polarization	T & R	✓	900
LAVA [73]	2.4 GHz	Amplitude	T	✓	/
ScatterMIMO [29]	5 GHz	Phase	R	✓	450
RFlens [31]	5 GHz	Phase	T	✓	246
Diffract [54]	5 GHz	Diffraction	T	✗	33
Scrolls [47]	0.9-6 GHz	Frequency	R	✓(row-wise)	156
mmWall [25]	24 GHz	Phase	T & R	✓(column-wise)	~10K
NR-Surface [37]	24 GHz	Phase	R	✓(column-wise)	600
PMSat [55]	20 & 30 GHz	Phase	T	✗	30
MilliMirror [57]	60 GHz	Phase	R	✗	15
AutoMS [48]	60 GHz	Phase	R	✗	<2

Table 1: Diverse hardware designs[1], transmissive (T) and reflective (R).

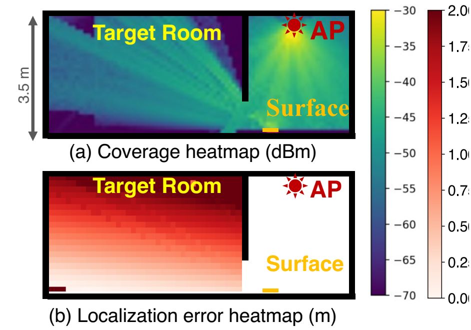


Figure 2: Lacking support for multiple services concurrently. A surface configuration to maximize coverage can disrupt localization.

Most work improves the perceived wireless channel conditions for better coverage and/or throughput [15, 29, 31, 40, 42, 43, 47, 66, 73]; Other work explores sensing applications [20, 74, 75, 77], security protection [41, 63], wireless powering/charging [14, 22, 64, 78] and more. For higher frequencies like 60 GHz, surfaces can effectively extend the link range [23, 25, 37, 48, 57] and sensing [68] by circumventing environmental blockage. Recent investigations also cover the security threats from metasurfaces [24, 61, 76]. Additionally, many RIS works focus on theoretical channel modeling or small-scale, single-link experimental validation [35]. [44, 45] propose a per-surface *tile* abstraction for network-layer modeling, but assume an oversimplified empty environment fully covered with surfaces and overlook crucial considerations for practical surfaces, such as surface control granularity and heterogeneity in the hardware designs and use cases.

2.1 The need for an OS

For surfaces to be integrated in future wireless networks, we are missing critical system support.

Heterogeneous hardware. There have been numerous surface designs (Table 1) and they are expected to diverge further to suit diverse use cases. First, controlling each fundamental signal property, e.g., phase, amplitude, and frequency, requires different element patterns and circuitry, with specific designs incorporating one or multiple control primitives. Second, any element design can only cater to a limited wavelength range, hence covering frequencies from sub-6 GHz to the millimeter wave (mmWave) range needs multiple designs. Third, a surface can fix its configurations during fabrication (passive, one-time programmable) or support dynamic reconfiguration (programmable). On high frequencies especially, due to the hardware complexity and high components costs, programmable surfaces [23, 25, 37] cost over $2 *per element*, and often only support *column-wise* reconfiguration (shared element states per column) rather than *element-wise* (distinct element states within a column). The cost easily reaches thousands of dollars for a large array of elements. In contrast, fully passive surfaces [48, 55, 57] are very low-cost, e.g., $1 for 60 thousand elements in AutoMS, and need no power. Fourth, deployment constraints may call for transmissive or reflective surfaces, rigid or flexible surface substrates [46, 52, 72]. Fifth, we may add new surfaces with updated designs over time incrementally. To manage all these surfaces across environments, we lack unified hardware programming interfaces to enable a universal control plane.

Diverse services. We refer to low-level capabilities or features (e.g., coverage, sensing, powering) as *services* from surfaces. While surface systems can operate in myriad ways, existing systems only support one service each, or two services [31, 77] without specifying how to handle both together. To juxtapose multiple services, we can deploy either multiple surfaces each for one service or one surface for multiple services, both strategies problematic. First, different surfaces can interfere with each other's operations. For example, surfaces designed for 2.4 GHz may block 3 GHz cellular and 5 GHz Wi-Fi signals, causing connectivity issues for other networks. Second, without explicit consideration for coexistence, different services can fail to share the same hardware Figure 2 shows an example of a surface for mmWave coverage extension from the AP to the target room. The surface configuration to optimize for coverage can disrupt or preclude effective user localization in the same space, since the surface operations can inadvertently invalidate spatial information assumptions for the localization algorithm. To solve this, we need a central control plane to efficiently utilize hardware and coordinate diverse services for multiprogramming.

User applications. SurfOS operates at the last hop, managing the environments where end users reside. While existing systems optimize for signal-level metrics like SNR or RSSI, this

[1] This is not an exhaustive list; Selected examples are sorted by theoperating frequency band, re-configurability, and publishing date.

does not always align with or efficiently fulfill the application-level end user demands. For example, video streaming favors smooth link conditions, but the SNR enhancement from surfaces may be insufficient, unstable, or excessive. Additionally, application demands vary. VR/AR gaming needs high throughput and low latency, smart home applications need sensing capability, while sensitive data transmission necessitates added security protection. We lack system support to translate user demands and invoke suitable surface services.

3 Orchestrating Surfaces for Services

SurfOS consists of three abstraction layers (Figure 3): *hardware manager*, *surface orchestrator*, and *service broker*. Together they aim to address three **challenges**:

Hardware heterogeneity and interactions. There have been numerous hardware designs due to wide-ranging surface capabilities and trade-offs (Table 1). Moreover, the shared wireless propagation medium leads to inherent and implicit interactions between hardware, which can be destructive or constructive. SurfOS needs well-defined APIs to mask hardware heterogeneity while capturing their interactions.

Service scheduling and multiplexing. Efficient multi-service provisioning requires careful hardware scheduling and task multiplexing in a shared radio environment. Typical OSes manage hardware that behaves in known and controllable manners. In contrast, surfaces cannot fully control the shared medium, which is susceptible to unknown and dynamic external events such as human movement. Thus, SurfOS needs to capture the environmental conditions through wireless channel simulations or endpoint feedback to orchestrate surfaces.

Application demand awareness. Given the unique role of surfaces – signal-level infrastructure serving ultimately user facing applications, SurfOS needs to ensure the signal-level objectives meet the application demands. Further, SurfOS should support both existing wireless applications and facilitate the development of new surface-native applications.

3.1 Hardware Manager

The hardware manager aims to hide hardware details from the upper layers with unified APIs.

Masking heterogeneity. SurfOS leverages *drivers* to mask the hardware details and expose unified APIs. Despite numerous hardware design possibilities, fortunately, the distinct signal properties are limited. Therefore, we propose abstractions corresponding to the fundamental signal properties – amplitude, phase, frequency, and polarization – for signal property control, like `set_amplitude()`, `shift_phase()`, ..., analogous to the `read()` and `write()` primitives for file systems. The input to these primitives is surface *configurations*. One configuration is an array of signal property alteration values for each surface element, e.g., phase shift values.

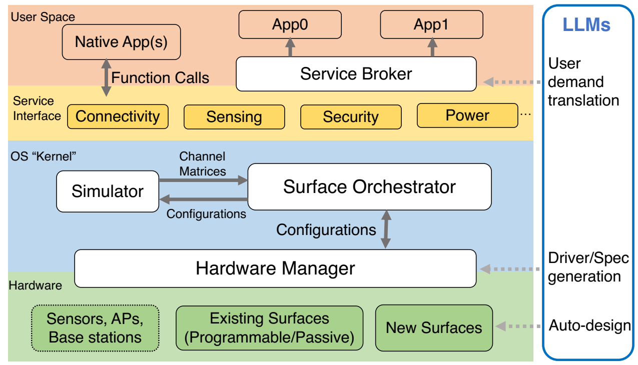


Figure 3: SurfOS Ecosystem Overview. We envision an OS-like "kernel" to manage surface hardware and provide programming interfaces for wireless services; These APIs lend to workflow automation using LLMs.

Programmable surfaces may accept multiple configurations to allow dynamic configuration switching. The primitives should provide the finest control granularity, e.g., element-wise surface configuration. This provides maximal flexibility for the upper layers to multiplex and compose surface control.

Decoupling surface management from dynamic adaption. To adapt surfaces to environmental changes or endpoint mobility, the control module often needs to react in real time, sometimes requiring feedback from communication endpoints. This sets a challenging latency budget, and could limit SurfOS to a local server rather than an edge or cloud server. Accordingly, SurfOS decouples general hardware management (control plane) from real-time signal-level actuation (data plane), similar to how SDN decouples the control plane from the data plane. Surface drivers manage surfaces by updating surfaces' *locally stored configurations*, analogous to forwarding tables on switches or beamforming codebooks for 802.11ad APs [48]. For example, a surface may store multiple sets of phase shift values as local configurations, each for a distinct beam direction. Based on the endpoint feedback, a surface reacts locally to choose the best configuration. When the upper layer reconfigures the surfaces, the configurations can be updated asynchronously through the drivers.

Hardware specifications. The drivers also explicitly capture and expose key hardware parameters to the upper layer for efficient hardware utilization. The specifications should allow the upper layer to model hardware behaviors correctly. We list representative parameters below: (i) Wideband frequency response, indicating reflection or transmission efficiency across the spectrum to avoid unintended blocking. (ii) Operation mode, whether this surface reflects signals or transmits the signals through, or both. (iii) Control delay, the delay for SurfOS to update configurations of a (remotely) controlled surface. For programmable surfaces, the control delay can vary from microseconds to milliseconds. Passive surfaces only have one-time configurability during fabrication, i.e., infinite control delay, similar to ROM versus other storage.

Non-surface hardware. SurfOS also manages or interacts with non-surface hardware, including sensors, APs, base stations. This allows us to obtain sensing input or channel feedback from end user devices to guide the surface reconfiguration and, when available, interact with the APs and base stations for better performance. For cellular and 802.11ad networks, such feedback mechanism is already part of the MAC protocol and can interface with SurfOS. Custom external sensors can directly report measurements to SurfOS, e.g., power detectors in LAVA, Lidar in AutoMS, or cameras and mmWave radars.

3.2 Surface Orchestrator

Service APIs. This layer exposes service request APIs to the upper layer, e.g., `enhance_link()`, `enable_sensing()`, and `init_powering()` for coverage, sensing, and powering services respectively. These are environment-wide abstractions, not associated with specific hardware, to mask the details of which or how surfaces provide the requested services. Each function call specifies the service goals as input and creates a *task* (akin to OS processes). Then, the surface orchestrator dynamically performs service tasks by scheduling all surface hardware globally – calling hardware manager APIs and setting surface states. It includes a scheduler for task scheduling and multiplexing and an optimizer to optimize surface configurations for multiple tasks. It also uses a channel simulator to model the interactions between surfaces.

Service scheduling. The scheduler should exploit task dynamics to optimize hardware utilization, i.e., setting a task idle when not used and releasing resources. This is conceptually similar to the conventional OS process scheduler, but the challenging research problem is how to provide modern OS features, such as priority support, performance guarantee, and task isolation, in the context of wireless networks.

Modeling interactions. Although the hardware manager provides surface programming interfaces, the shared wireless medium can cause hidden correlations and break their independence. The deployment environment also impacts surface operations. To address this, we use a wireless channel simulator to model such interactions among surfaces and the environment. This modeling facilitates collaboration and avoids interference between surfaces. Given surface specifications and the 3D environment model as input, the simulator outputs the channel matrices between the surfaces and endpoints on the relevant frequency bands. The surface orchestrator uses these channel matrices to calculate service performance metrics, such as the received signal strength and estimated sensing or localization accuracy. This lays the foundation for the following task multiplexing and optimization.

Task multiplexing. Similar to handling digital signal transmission, we consider several dimensions for multiplexing: time, frequency, and space. Depending on the actuation speed, the surfaces can switch between configurations for different tasks to achieve time division multiplexing; Surfaces may operate multiple tasks on distinct frequency bands simultaneously to achieve frequency division multiplexing; A large surface or distributed surfaces can adopt space division multiplexing, i.e., spatially grouped by tasks, according to proximity to or channel response strengths at targeted devices. Thus, the minimal resource scheduling unit assigned to a task would be a slice of time, frequency, and space.

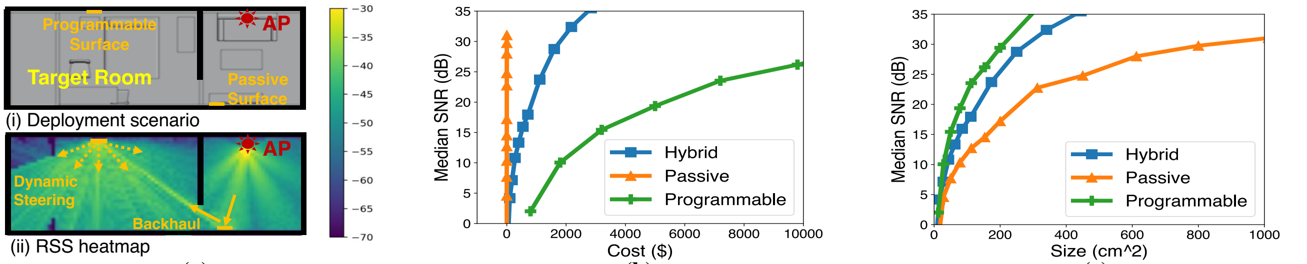
Configuration optimization. Given the scheduled tasks, we use an optimizer to achieve the desired service performance. Based on the channel modeling by the wireless simulator, an optimizer searches the surface configurations for multiples surfaces so that surfaces can collaborate based on the unified hardware manager APIs.

Multitasking with joint optimization. We highlight a new opportunity of surface multitasking, i.e., multiple concurrent tasks or services can use the same hardware resource unit without conflicts, such as joint communication and sensing. This is a new dimension for multiplexing – surface-wide configuration multiplexing – analogous to code-division multiplexing. Intuitively, the surface can steer the main beam towards certain regions for coverage, while the side lobes can be used for sensing. SurfOS does not dictate a fixed set of codes for each task, but uses the aforementioned optimizer to search for the configurations.

3.3 Service Broker

By masking hardware heterogeneity and provide service abstractions, we can simplify development of applications that utilize surfaces. For existing applications not aware of surfaces, we introduce a service broker, as a base application (a daemon), that invoke services based on their demands.

Instead of focusing on signal-level metrics as in prior work, the service broker should call surface services according to application-level demands of end users. It is challenging to translate user demands or application performance targets to low-level service targets for surfaces, e.g., signal-level metrics. For example, translating guaranteed VR experience to SNR improvement involves multiple non-linear mappings across network stack layers. User intent and habits come into play, affecting which service to invoke or end. Additionally, a single application may involve multiple services, hence requiring multi-objective optimization. We can potentially sense or monitor wireless traffic to understand user demands. A promising solution is to use LLMs, as discussed below.



(a) (b) (c)

Figure 4: Leveraging hardware heterogeneity. (a.i) Experiment setup. **(a.ii)** Surface collaboration: The passive surface serves as a narrow-beam backhaul, while a programmable surface re-steers the beam dynamically to cover the room. A hybrid solution can flexibly balance multiple trade-offs among surface cost **(b)**, size **(c)**, re-configurability, and deployment complexity.

3.4 LLMs for Automation

The abstraction layers above present clear APIs that ease workflow automation by facilitating code generation and module synthesis. Taking a leaf out of recent proposals of using LLMs for network management [49, 51, 62], SurfOS uses LLMs as an external tool to (1) translate user intent into service function calls, and (2) generate surface hardware driver code. This reduces the manual efforts for driver and application development, which can encourage system adoption.

User demand translation. As LLMs become integral to AI assistants on personal devices [3, 11, 13], they serve as the primary user interface to invoke applications and act upon user intent. Naturally, they can also serve as assistants or administrators to manage the radio environments and surface services. By interpreting user intent described in natural languages, LLMs can invoke surface applications or directly generate new surface applications that call for services from SurfOS.

Hardware driver generation. The diversity of existing surface designs necessitates substantial effort to document their specifications and develop drivers in a unified format. LLMs can assist by parsing and summarizing long text, such as datasheets or research papers, to generate surface hardware specifications, similar to extracting protocol specifications in prior work [62]. On that basis, LLMs may further synthesize the driver code based on the specifications generated.

4 Exploratory studies

In this section, we highlight the feasibility and benefits of our approach with an early-stage implementation. Besides typical OS benefits, e.g., easy hardware management and reusable software modules, we explore two important functionalities of SurfOS via simulation: managing heterogeneous surfaces with unified configurations (hardware manager) and task multiplexing via joint optimization (surface orchestrator).

Implementation. We first implement a unified configuration interface of the hardware manager for phase-control metasurfaces. Specifically, a passive surface takes a single set of per-element phase shift values, while each programmable surface takes multiple sets of element-wise states. The best set for a programmable surface is chosen based on endpoint feedback, following recent work [25, 37]. We further implement the optimizer of the surface orchestrator, which jointly or individually optimizes for coverage and localization services with surface configurations as variables. The optimizer uses gradient descent, while other algorithms can be easily supported. Lastly, we use the open-source AutoMS channel simulator [1] to model channel conditions for SurfOS. It supports high-accuracy channel modeling, validated experimentally, of large-sized metasurfaces. To evaluate SurfOS performance, we calculate link SNR as in AutoMS [48] and estimate AoA (angle-of-arrival) according to md-Track [18, 70]. The AoA between the client device and metasurface is estimated based on the channel information from the AP, then converted to localization error assuming accurate ToF (time of flight).

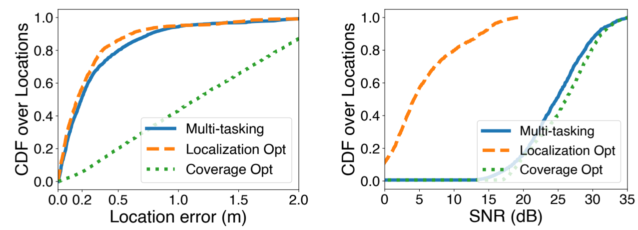


Figure 5: Multitasking for joint localization and coverage. Joint optimization ensures high performance for both tasks with a single surface configuration.

Heterogeneous surface management. We use two rooms of a furnished apartment as our testing scenario (Figure 4a). An AP is placed near the living room wall, and we want to extend mmWave coverage to the adjacent bedroom. We deploy two surfaces, one passive and one programmable, at suitable pre-determined deployment locations. They represent two extremes of the design spectrum: Passive surfaces incur extreme low cost, zero power, and easy setup, but need a much larger hardware area size that may not fit at many deployment

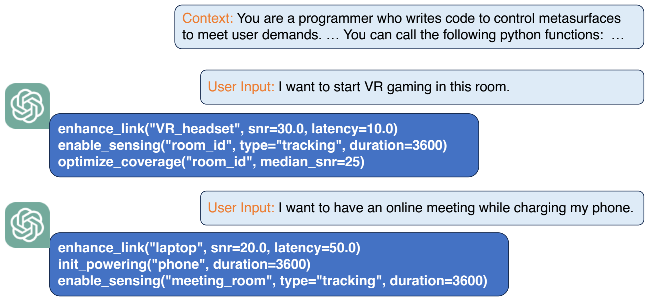


Figure 6: LLM calling surface services. Explanatory code comments from LLM are omitted.

locations; programmable surfaces impose a smaller spatial footprint via re-configurability, but are far more expensive and complex. We combine them into *a hybrid solution* to explore flexible trade-offs. The optimizer optimizes the configurations for both surfaces simultaneously and effortlessly, owing to the unified surface configuration interface. Figure 4 shows the cost and sizes needed to reach different median SNRs in the target room. Without surfaces, there is basically no coverage in the target room, so the median SNRs also represent the SNR *gains*. Compared with passive-only and programmable-only approaches, our hybrid solution only needs a fraction of the hardware cost and size for comparable performance. Intuitively, we exploit the individual advantages of the two designs simultaneously, by using the passive surface as a backhaul to relay the beam and using the programmable surface to dynamically steer the beam for coverage.

Surface multitasking. To explore multitasking, we optimize for localization and coverage task performance jointly with a shared surface configuration, following the setup in Figure 2. We define the loss function of the localization task as the cross-entropy between the estimated and true AoA, and the loss function of the coverage task as the negative sum of link capacity across different locations. With the phase shift values (surface configuration) as variables, we minimize the sum of localization loss and coverage loss. We compare the performance of our multitasking configuration with single-tasking configurations optimized for localization and coverage individually. Figure 5 shows CDF of performance across locations inside the target room. We note that a single surface configuration can effectively multitask with little performance loss, saving hardware resources significantly. Although we consider a passive surface here, programmable surfaces can perform similar multitasking within each time slot.

Translating user demands. We further show abbreviated input and output of GPT-4o [12] (Figure 6) to demonstrate how LLMs can help translate user demands to SurfOS function calls. The LLM is able to request services from surfaces based on the user's natural language input.

5 Discussion

OS versus libraries or SDKs. While it may appear that libraries for surface control is an alternative to SurfOS, we argue that an OS-like runtime is necessary. Libraries or SDKs provide compile-time support, while we need run-time support to handle the dynamics of the radio environments. As surfaces can not completely control a physical environment, events such as furniture movement and people walking can require dynamic reconfiguration of surface states. Further, nearby wireless endpoints can request services with distinct demands. A runtime system can monitor and coordinate surface hardware to adapt to varying wireless channel changes and application demands.

Native application development. With SurfOS, new surface-native applications can be developed by directly invoking and combining surface services. Moreover, similar to how SDN paves the way for new services, such as network function virtualization (NFV) and virtualization of RAN (vRAN), the centralized control plane of SurfOS can enable new features, such as network monitoring, diagnosis, and wireless propagation environment virtualization.

New hardware design and deployment. Surface systems encompass three key stages: design, deployment, and management. Our discussion so far has assumed the first two stages to be predetermined and focused on managing existing surfaces. However, in clean slate scenarios, we also need to consider the design and deployment stages. Existing systems often rely on expert knowledge to determine suitable designs (e.g., element patterns, components, substrate materials) and deployment setups (e.g., surface locations, sizes). AutoMS [48] is the first to explore workflow automation including the hardware design and placement, but specifically for mmWave coverage with passive surfaces. The abstraction layers of SurfOS make it easy to streamline and automate the entire process for generalized hardware types and use cases. This involves compiling upper-layer goals into hardware designs and deployment configurations. For design automation, based on the user input, LLMs can locate an appropriate design from a surface design database. If existing designs are inadequate, for instance, when a new operating frequency band is needed, LLMs can determine the necessary design parameter adjustments, then initiate electromagnetic simulation software to optimize and produce a new design. Deployment automation involves running the simulator to model the environment and optimize for placement as part of the surface hardware configurations.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions. This work is partially funded by the National Science Foundation under Grant No. 2112562.

References

[1] [n. d.]. GitHub repository of AutoMS. https://github.com/microsoft/AutoMS.

[2] [n. d.]. Greener Wave, French metasurface startup. https://greenerwave.com/our-technology/,.

[3] [n. d.]. Microsoft Copilot. https://www.microsoft.com/en-us/microsoft-copilot,.

[4] [n. d.]. NEC smart surface. https://www.nec-enterprise.com/it/projects,.

[5] 2022. EMPOWER: Final technology roadmap for advanced wireless. https://www.advancedwireless.eu/wp-content/uploads/2022/02/Del2-5-Final-technology-roadmap.pdf,.

[6] 2022. Next G Alliance Report: Roadmap to 6G. https://nextgalliance.org/wp-content/uploads/2022/02/NextGA-Roadmap.pdf,.

[7] 2023. China Mobile and ZTE successfully complete application verification of 5G-A Reconfigurable Intelligent Surface for Asian Games. https://www.zte.com.cn/global/about/news/china-mobile-and-zte-successfully-complete-application-verification-of-5g-a-reconfigurable-intelligent-surface-ris-for-asian-games.html.

[8] 2023. DOCOMO Conducts World's First Trial of Transmissive Metasurface on Window to Deliver Indoor Radio Waves to Outdoor Foot of Building. https://www.docomo.ne.jp/english/info/media_center/pr/2023/0130_02.html,.

[9] 2023. ETSI first report for RIS standardization. https://www.etsi.org/deliver/etsi_gr/RIS/001_099/001/01.01.01_60/gr_RIS001v010101p.pdf,.

[10] 2023. ITU International Mobile Telecommunications (IMT) for 2030 and beyond. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2160-0-202311-I!!PDF-E.pdf,.

[11] 2024. Apple Intelligence for iPhone, iPad, and Mac. https://www.apple.com/newsroom/2024/06/introducing-apple-intelligence-for-iphone-ipad-and-mac/,.

[12] 2024. GPT-4o. https://openai.com/index/hello-gpt-4o/,.

[13] 2024. Meta Smart Glasses with Meta AI. https://www.meta.com/smart-glasses/,.

[14] Devansh R Agrawal, Yuji Tanabe, Desen Weng, Andrew Ma, Stephanie Hsu, Song-Yan Liao, Zhe Zhen, Zi-Yi Zhu, Chuanbowen Sun, Zhenya Dong, et al. 2017. Conformal phased surfaces for wireless powering of bioelectronic microdevices. *Nature biomedical engineering* 1, 3 (2017), 0043.

[15] Venkat Arun and Hari Balakrishnan. 2020. RFocus: Practical Beamforming for Small Devices. In *Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 1047–1061.

[16] Manu Bansal, Jeffrey Mehlman, Sachin Katti, and Philip Levis. 2012. Openradio: a programmable wireless dataplane. In *Proceedings of the first workshop on Hot topics in software defined networks*. 109–114.

[17] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M. Alouini, and R. Zhang. 2019. Wireless Communications Through Reconfigurable Intelligent Surfaces. *IEEE Access* 7 (2019).

[18] Alejandro Blanco, Pablo Jiménez Mateo, Francesco Gringoli, and Joerg Widmer. 2022. Augmenting mmWave localization accuracy through sub-6 GHz on off-the-shelf devices. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. 477–490.

[19] Martin Casado, Michael J Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker. 2007. Ethane: Taking control of the enterprise. *ACM SIGCOMM computer communication review* 37, 4 (2007), 1–12.

[20] Baicheng Chen, John Nolan, and Xinyu Zhang. 2024. MetaBioLiq: A Wearable Passive Metasurface Aided mmWave Sensing Platform for BioFluids. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[21] Lili Chen, Wenjun Hu, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Jeremy Gummeson. 2021. Pushing the physical limits of IoT devices with programmable metasurfaces. In *Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 425–438.

[22] Lili Chen, Bozhong Yu, Yongjian Fu, Ju Ren, Jeremy Gummeson, and Yaoxue Zhang. 2024. Pushing Wireless Charging from Station to Travel. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[23] Lili Chen, Bozhong Yu, Ju Ren, Jeremy Gummeson, and Yaoxue Zhang. 2023. Towards Seamless Wireless Link Connection. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. 137–149.

[24] Xingyu Chen, Zhengxiong Li, Biacheng Chen, Yi Zhu, Chris Xiaoxuan Lu, Zhengyu Peng, Feng Lin, Wenyao Xu, Kui Ren, and Chunming Qiao. 2023. Metawave: Attacking mmwave sensing with meta-material-enhanced tags. In *The 30th Network and Distributed System Security (NDSS) Symposium 2023*. The Internet Society, 1–17.

[25] Kun Woo Cho, Mohammad H Mazaheri, Jeremy Gummeson, Omid Abari, and Kyle Jamieson. 2023. mmWall: A Steerable, Transflective Metamaterial Surface for NextG mmWave Networks. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1647–1665.

[26] Tie Jun Cui, Mei Qing Qi, Xiang Wan, Jie Zhao, and Qiang Cheng. 2014. Coding metamaterials, digital metamaterials and programmable metamaterials. *Light: science & applications* 3, 10 (2014), e218–e218.

[27] Marco Di Renzo, Alessio Zappone, Merouane Debbah, Mohamed-Slim Alouini, Chau Yuen, Julien De Rosny, and Sergei Tretyakov. 2020. Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead. *IEEE journal on selected areas in communications* 38, 11 (2020), 2450–2525.

[28] Colin Dixon, Ratul Mahajan, Sharad Agarwal, AJ Brush, Bongshin Lee, Stefan Saroiu, and Paramvir Bahl. 2012. An operating system for the home. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*. 337–352.

[29] Manideep Dunna, Chi Zhang, Daniel Sievenpiper, and Dinesh Bharadia. 2020. ScatterMIMO: enabling virtual MIMO with smart surfaces. In *Proceedings of International Conference on Mobile Computing and Networking (Mobicom)*. ACM, 1–14.

[30] Nick Feamster, Jennifer Rexford, and Ellen Zegura. 2014. The road to SDN: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review* 44, 2 (2014), 87–98.

[31] Chao Feng, Xinyi Li, Yangfan Zhang, Xiaojing Wang, Liqiong Chang, Fuwei Wang, Xinyu Zhang, and Xiaojiang Chen. 2021. RFlens: metasurface-enabled beamforming for IoT communication and sensing. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (MobiCom '21)*. ACM, New York, NY, USA, 587–600.

[32] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. 2008. NOX: towards an operating system for networks. *ACM SIGCOMM computer communication review* 38, 3 (2008), 105–110.

[33] Aditya Gudipati, Daniel Perry, Li Erran Li, and Sachin Katti. 2013. SoftRAN: Software defined radio access network. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. 25–30.

[34] Chi-Yao Hong, Subhasree Mandal, Mohammad Al-Fares, Min Zhu, Richard Alimi, Chandan Bhagat, Sourabh Jain, Jay Kaimal, Shiyu Liang, Kirill Mendelev, et al. 2018. B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN. In *Proceedings of the 2018 Conference of the*

ACM Special Interest Group on Data Communication. 74–87.

[35] Jie Huang, Cheng-Xiang Wang, Yingzhuo Sun, Rui Feng, Jialing Huang, Bolun Guo, Zhimeng Zhong, and Tie Jun Cui. 2022. Reconfigurable intelligent surfaces: Channel characterization and modeling. *Proc. IEEE* 110, 9 (2022), 1290–1311.

[36] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. 2013. B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 3–14.

[37] Minseok Kim, Namjo Ahn, and Song Min Kim. 2024. {NR-Surface}:{NextG-ready}{μW-reconfigurable}{mmWave} Metasurface. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*.

[38] Teemu Koponen, Martin Casado, Natasha Gude, Jeremy Stribling, Leon Poutievski, Min Zhu, Rajiv Ramanathan, Yuichiro Iwata, Hiroaki Inoue, Takayuki Hama, et al. 2010. Onix: A distributed control platform for large-scale production networks. In *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*.

[39] Geoffroy Lerosey and Mathias Fink. 2022. Wavefront shaping for wireless communications in complex media: From time reversal to reconfigurable intelligent surfaces. *Proc. IEEE* 110, 9 (2022), 1210–1226.

[40] Ruinan Li, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2024. Plug-and-play Indoor GPS Positioning System with the Assistance of Optically Transparent Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[41] Xinyi Li, Chao Feng, Fengyi Song, Chenghan Jiang, Yangfan Zhang, Ke Li, Xinyu Zhang, and Xiaojiang Chen. 2022. Protego: securing wireless communication via programmable metasurface. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 55–68.

[42] Xinyi Li, Gaoteng Zhao, Ling Chen, Xinyu Zhang, and Ju Ren. 2024. RFMagus: Programming the Radio Environment With Networked Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[43] Zhuqi Li, Yaxiong Xie, Longfei Shangguan, Rotman Ivan Zelaya, Jeremy Gummeson, Wenjun Hu, and Kyle Jamieson. 2019. Towards Programming the Radio Environment with Large Arrays of Inexpensive Antennas. In *Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 285–300.

[44] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I Akyildiz. 2018. A New Wireless Communication Paradigm through Software-Controlled Metasurfaces. *IEEE Communications Magazine* 56, 9 (2018).

[45] Christos Liaskos, Ageliki Tsioliaridou, Shuai Nie, Andreas Pitsillides, Sotiris Ioannidis, and Ian F Akyildiz. 2019. On the network-layer modeling and configuration of programmable wireless environments. *IEEE/ACM transactions on networking* 27, 4 (2019), 1696–1713.

[46] Ruichun Ma and Wenjun Hu. 2024. RF-Mediator: Tuning Medium Interfaces with Flexible Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[47] Ruichun Ma, R. Ivan Zelaya, and Wenjun Hu. 2023. Softly, Deftly, Scrolls Unfurl Their Splendor: Rolling Flexible Surfaces for Wideband Wireless. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23)*. ACM, New York, NY, USA, Article 18, 15 pages.

[48] Ruichun Ma, Shicheng Zheng, Hao Pan, Lili Qiu, Xingyu Chen, Liangyu Liu, Yihong Liu, Wenjun Hu, and Ju Ren. 2024. AutoMS: Automated Service for mmWave Coverage Optimization using Low-cost Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[49] Sathiya Kumaran Mani, Yajie Zhou, Kevin Hsieh, Santiago Segarra, Trevor Eberl, Eliran Azulai, Ido Frizler, Ranveer Chandra, and Srikanth Kandula. 2023. Enhancing network management using code generated by large language models. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*. 196–204.

[50] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review* 38, 2 (2008), 69–74.

[51] Rajdeep Mondal, Alan Tang, Ryan Beckett, Todd Millstein, and George Varghese. 2023. What do LLMs need to Synthesize Correct Router Configurations?. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*. 189–195.

[52] Placido Mursia, Francesco Devoti, Marco Rossanese, Vincenzo Sciancalepore, Gabriele Gradoni, Marco Di Renzo, and Xavier Costa-Perez. 2024. T3DRIS: Advancing Conformal RIS Design through In-depth Analysis of Mutual Coupling Effects. *arXiv preprint arXiv:2404.05261* (2024).

[53] RCR Wireless News. 2024. Qualcomm focused on making 5G mmWave more cost effective. https://www.rcrwireless.com/20240325/spectrum/qualcomm-focused-on-making-5g-mmwave-more-cost-effective,.

[54] Anurag Pallaprolu, Winston Hurst, Sophia Paul, and Yasamin Mostofi. 2023. I Beg to Diffract: RF Field Programming With Edges. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[55] Hao Pan, Lili Qiu, Bei Ouyang, Shicheng Zheng, Yongzhao Zhang, Yi-Chao Chen, and Guangtao Xue. 2023. PMSat: Optimizing Passive Metasurface for Low Earth Orbit Satellite Communication. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[56] Carl Pfeiffer and Anthony Grbic. 2013. Metamaterial Huygens' surfaces: tailoring wave fronts with reflectionless sheets. *Physical review letters* 110, 19 (2013), 197401.

[57] Kun Qian, Lulu Yao, Xinyu Zhang, and Tse Nga Ng. 2022. MilliMirror: 3D printed reflecting surface for millimeter-wave coverage expansion. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 15–28.

[58] Marco Di Renzo, Merouane Debbah, Dinh-Thuy Phan-Huy, Alessio Zappone, Mohamed-Slim Alouini, Chau Yuen, Vincenzo Sciancalepore, George C Alexandropoulos, Jakob Hoydis, Haris Gacanin, et al. 2019. Smart radio environments empowered by reconfigurable AI metasurfaces: An idea whose time has come. *EURASIP Journal on Wireless Communications and Networking* 2019, 1 (2019), 1–20.

[59] Julius Schulz-Zander, Carlos Mayer, Bogdan Ciobotaru, Stefan Schmid, and Anja Feldmann. 2015. OpenSDWN: Programmatic control over home and enterprise WiFi. In *Proceedings of the 1st ACM SIGCOMM symposium on software defined networking research*. 1–12.

[60] Julius Schulz-Zander, Lalith Suresh, Nadi Sarrar, Anja Feldmann, Thomas Hühn, and Ruben Merz. 2014. Programmatic Orchestration of {WiFi} Networks. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*. 347–358.

[61] Zhambyl Shaikhanov, Sherif Badran, Hichem Guerboukha, Josep Jornet, Daniel Mittleman, and Edward Knightly. 2024. MetaFly: Wireless Backhaul Interception via Aerial Wavefront Manipulation. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 151–151.

[62] Prakhar Sharma and Vinod Yegneswaran. 2023. PROSPER: Extracting Protocol Specifications Using Large Language Models. In *Proceedings*

of the 22nd ACM Workshop on Hot Topics in Networks. 41–47.

[63] Jayanth Shenoy, Zikun Liu, Bill Tao, Zachary Kabelac, and Deepak Vasisht. 2022. RF-protect: privacy against device-free human tracking. In *Proceedings of the ACM SIGCOMM 2022 Conference*. 588–600.

[64] Yiwen Song, Hao Pan, Longyuan Ge, Lili Qiu, Swarun Kumar, and Yi-Chao Chen. 2024. MicroSurf: Guiding Energy Distribution inside Microwave Oven with Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[65] Mikko A Uusitalo, Patrik Rugeland, Mauro Renato Boldi, Emilio Calvanese Strinati, Panagiotis Demestichas, Mårten Ericson, Gerhard P Fettweis, Miltiadis C Filippou, Azeddine Gati, Marie-Helene Hamon, et al. 2021. 6G Vision, Value, Use Cases and Technologies From European 6G Flagship Project Hexa-X. *IEEE Access* 9 (2021), 160004–160020.

[66] Yezhou Wang, Hao Pan, Lili Qiu, Linghui Zhong, Jiting Liu, Ruichun Ma, Yi-Chao Chen, Guangtao Xue, and Ju Ren. 2024. GPMS: Enabling Indoor GNSS Positioning using Passive Metasurfaces *(ACM MobiCom '24)*. ACM, New York, NY, USA.

[67] Allen Welkie, Longfei Shangguan, Jeremy Gummeson, Wenjun Hu, and Kyle Jamieson. 2017. Programmable Radio Environments for Smart Spaces. In *Workshop on Hot Topics in Networks (HotNets)*. ACM.

[68] Timothy Woodford, Kun Qian, and Xinyu Zhang. 2023. Metasight: High-Resolution NLoS Radar Sensing through Efficient Metasurface Encoding. In *Proceedings of the 21th ACM Conference on Embedded Networked Sensor Systems*.

[69] Q. Wu and R. Zhang. 2020. Towards Smart and Reconfigurable Environment: Intelligent Reflecting Surface Aided Wireless Network. *IEEE Communications Magazine* 58, 1 (2020).

[70] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. 2019. mD-Track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[71] Jiarong Xing, Yiming Qiu, Kuo-Feng Hsu, Hongyi Liu, Matty Kadosh, Alan Lo, Aditya Akella, Thomas Anderson, Arvind Krishnamurthy, T. S. Eugene Ng, and Ang Chen. 2021. A Vision for Runtime Programmable Networks. In *Proceedings of the 20th ACM Workshop on Hot Topics in Networks*. ACM, New York, NY, USA, 91–98. https://doi.org/10.1145/3484266.3487377

[72] R Ivan Zelaya, Ruichun Ma, and Wenjun Hu. 2021. Towards 6G and Beyond: Smarten Everything with Metamorphic Surfaces. In *Proceedings of the Twentieth ACM Workshop on Hot Topics in Networks*. 155–162.

[73] R Ivan Zelaya, William Sussman, Jeremy Gummeson, Kyle Jamieson, and Wenjun Hu. 2021. LAVA: fine-grained 3D indoor wireless coverage for small IoT devices. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*. ACM, 123–136.

[74] Ganlin Zhang, Dongheng Zhang, Hongyu Deng, Yun Wu, Fengquan Zhan, and Yan Chen. 2024. Practical Passive Indoor Localization With Intelligent Reflecting Surface. *IEEE Transactions on Mobile Computing* (2024).

[75] Hongliang Zhang, Boya Di, Kaigui Bian, Zhu Han, H Vincent Poor, and Lingyang Song. 2022. Toward ubiquitous sensing and localization with reconfigurable intelligent surfaces. *Proc. IEEE* 110, 9 (2022), 1401–1422.

[76] Yangfan Zhang, Zhihao Hui, Hao Jia, Xiaojiang Chen, and Yaxiong Xie. 2024. Hydra: Attacking OFDM-base Communication System via Metasurfaces Generated Frequency Harmonics. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. ACM, New York, NY, USA.

[77] Yongzhao Zhang, Yezhou Wang, Lanqing Yang, Mei Wang, Yi-Chao Chen, Lili Qiu, Yihong Liu, Guangtao Xue, and Jiadi Yu. 2023. Acoustic Sensing and Communication Using Metasurface. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1359–1374.

[78] Jiafeng Zhou, Pei Zhang, Jiaqi Han, Long Li, and Yi Huang. 2021. Metamaterials and metasurfaces for wireless power transfer and energy harvesting. *Proc. IEEE* 110, 1 (2021), 31–55.