INVITED PAPER

Software-Defined Reconfigurable Intelligent Surfaces: From Theory to End-to-End Implementation

By Christos Liaskos, Lefteris Mamatas, Arash Pourdamghani, Ageliki Tsioliaridou, Sotiris Ioannidis, Andreas Pitsillides, *Senior Member IEEE*, Stefan Schmid, and Ian F. Akyildiz, *Life Fellow IEEE*

ABSTRACT | Programmable wireless environments (PWEs) utilize internetworked intelligent metasurfaces to transform wireless propagation into a software-controlled resource. In this article, the interplay is explored between the user devices, the metasurfaces, and the PWE control system from the theory to the end-to-end implementation. This article first discusses the metasurface hardware and software, covering the complete workflow from the user device initialization to its final service via the PWE. Furthermore, to be compatible with the 5G and 6G wireless systems, the software-defined networking (SDN) paradigm is extended to achieve scalable internetworking and central control in PWE deployments with multiple metasurfaces and multihop communication. Subsequently, the set of SDN foundations is exploited in order to abstract the physics behind PWEs and a theoretical framework is established to describe and manipulate them in an algorithmic form. This can lead to smart radio environments that are readily accessible from various engineering disciplines, facilitating their integration into existing networks, wireless systems, and applications. This article is concluded by outlining strategies for the optimal placement of metasurfaces within a PWE-controlled space, open challenges in PWE security, specialized SDN integration issues, and theoretical problems toward the graph-driven modeling of PWEs.

KEYWORDS | 6G; deployment optimization; HyperSurfaces; intelligent metasurfaces; internetworking; multihop; programmable wireless environments (PWEs); protocol stack; reconfigurable intelligent surfaces (RISs); security; smart radio environments (SREs); software-defined networking (SDN); theoretical foundations.

Manuscript received September 29, 2021; revised January 25, 2022; accepted April 17, 2022. This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme-Project C4IIoT under Grant EU833828 and in part by the European Research Council (ERC) under Grant Agreement 864228 (AdjustNet) (2020–2025). *(Corresponding author: Christos Liaskos.)*

Christos Liaskos is with the Computer Science and Engineering Department, University of Ioannina, GR45445 Ioannina, Greece, and also with the Foundation for Research and Technology Hellas (FORTH), GR70013 Heraklion, Greece (e-mail: cliaskos@uoi.gr).

Lefteris Mamatas is with the Department of Applied Informatics, University of Macedonia, GR54636 Thessaloniki, Greece (e-mail: emamatas@uom.edu.gr).

Arash Pourdamghani is with the Institute of Technology (TU Berlin), 10623 Berlin, Germany (e-mail: pourdamghani@tu-berlin.de).

Ageliki Tsioliaridou is with the Foundation for Research and Technology Hellas (FORTH), GR70013 Heraklion, Greece (e-mail: atsiolia@ics.forth.gr).

Sotiris Ioannidis is with the School of Electrical and Computer Engineering, Technical University of Crete, 731 00 Chania, Greece, and also with the Foundation for Research and Technology Hellas (FORTH), GR70013 Heraklion, Greece (e-mail: sotiris@ece.tuc.gr).

Andreas Pitsillides is with the Computer Science Department, University of Cyprus, 1678 Nicosia, Cyprus, and also with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, 2092 Johannesburg, South Africa (e-mail: andreas.pitsillides@ucy.ac.cy).

Stefan Schmid is with the Berlin Institute of Technology (TU Berlin), 10623 Berlin, Germany, and also with the Faculty of Computer Science, University of Vienna, 1010 Vienna, Austria (e-mail: stefan_schmid@univie.ac.at).

Ian F. Akyildiz is with Truva Inc., Alpharetta, GA 30022 USA (e-mail: ian@truvainc.com).

Digital Object Identifier 10.1109/JPROC.2022.3169917

I. INTRODUCTION

Recently, researchers have started envisioning innovative communication systems that rely on the control of the propagation of wireless signals in a 3-D space [1], [2]. This direction has become known under different nomenclatures: the *smart radio environments* (SREs), which focus on the novel signal processing capabilities and channel modeling, and the *programmable wireless environments* (PWEs), which treat the wireless propagation phenomenon as a deterministic, software-defined end-to-end service. Both approaches assume that planar objects within a 3-D space, such as walls and ceilings in an indoors' setting, are coated with a special material that can sense waves impinging upon them and alter them via electromagnetic (EM) *functions*. Exemplary functions, which include modifying the power, direction, phase, and polarization of a wireless wave [3], have been extensively studied and experimentally demonstrated [4]–[8].

The special materials employed in SREs and PWEs are metasurfaces (including exotic variants based on graphene for THz operation), as well as conceptually related technologies, such as phased antenna arrays and reflectarrays [1], [9], [10]. Each of these material types comprises a set of controllably radiating elements arranged over a 2-D layout (or even 3-D volumes/stack-ups). Each technology comes with a range of supported functions and efficiency degrees. Metasurfaces are such artificial materials with a very high density of radiating elements, containing 10–100 of these elements over a single wavelength of physical size [3]. Due to this high density, metasurfaces operate as transformers of the surface current distributions, i.e., created upon them by impinging waves. In theory, they are able to form any surface current distribution over them, thereby producing any EM output due to the Huygens principle [11], as illustrated in Fig. 1.

In this manner, highly efficient EM functions can be attained, even in the near field. Phased antenna arrays, also known as large intelligent surfaces (LISs), reconfigurable/reflective intelligent surfaces (RIS/IRS), and reflectarrays [10], [12], [13], are panels comprising a number of planar antennas with $(1/2)$ or $(1/4)$ wavelength separation in a 2-D grid layout. Each planar antenna is connected to a phase and amplitude shifter, such as p-i-n diodes or varactors/varistors. Using principles of the Fourier analysis, a required *far-field* radiation pattern of the reflectarray as a whole can be composed by proper phase and amplitude shifts applied per planar antenna to the impinging sinusoid signal [14]. In addition to the above solutions, the recently demonstrated HyperSurfaces constitute a type of internetworked metasurfaces [15], [16]. They comprise a software programming interface [application programming interface (API)], an *EM middleware*, and a gateway [5]. The API enables a systems engineer to read and alter EM functionality performed by the HyperSurface in real time while abstracting the underlying physics. The EM middleware translates API callbacks to corresponding active element states via a codebook approach. The gateway provides mainstream connectivity to existing network infrastructure (see Fig. 1).

A PWE is created by coating planar objects—such as walls and ceilings in an indoor environment—with

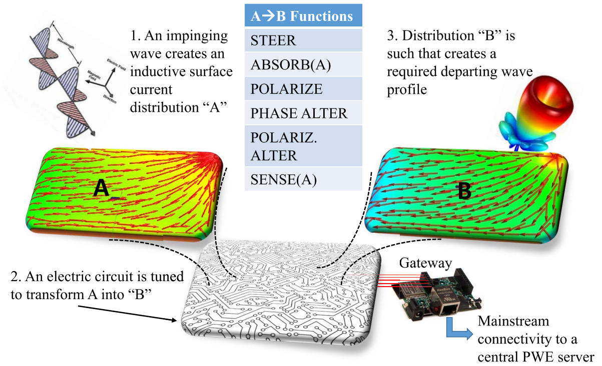


Fig. 1. ***Operating principle of intelligent surfaces based on the generalized internetworked metasurface concept (HyperSurface).***

intelligent metasurface tiles, i.e., stand-alone, rectangular, and individually addressable panels of any aforementioned technology, but with internetworking capabilities [17]. The latter allows a central server to connect to any tile, read its state, and deploy a new EM function in real time [2].

This article builds upon the existing literature and proposes a complete, end-to-end system model for PWEs, covering all aspects of hardware, software, control approaches, protocols, and theoretical foundations. The contributed system allows the operator to handle multiple wireless users with particular performance objectives or requirements, e.g., multiple objectives per user, user mobility, multicast groups, and partially coated PWEs that are supported. The objectives include power transfer and signal-to-interference maximization, as well as eavesdropping and Doppler effect mitigation.

Moreover, the study provides a meticulous integration strategy of PWEs into existing communications infrastructure by aligning them with the software-defined networking (SDN) principles [18]–[20]. The previously undefined workflow, initiating with the entry of a user device into the PWE vicinity and completing with its service and wireless channel customization, is now clarified. Most importantly, the study culminates with a foundational abstraction model that turns the PWE orchestration problem into a graph analysis equivalent. This is expected to allow software developers and engineers at large to develop novel PWE solutions without requiring background knowledge of physics, bringing PWEs closer to massive real-world applicability.

In order to achieve these goals, this article contributes the following:

1) a survey of metasurface hardware and software, covering the complete workflow from the user device initialization to its final service via the PWE;
2) an SDN paradigm to achieve scalable networking and central control in PWE deployments with multiple metasurfaces and multihop communication;
3) a theoretical framework for describing and manipulating PWE in a purely algorithmic form, while abstracting the physics behind them.

Furthermore, employing PWE simulation in specially designed tools, the study provides important insights into the problem of optimal placement of intelligent surfaces within a space. Finally, open challenges are highlighted in the areas of security, SDN integration, and theoretical analysis.

The remainder of this work is organized as follows. Section II provides a high-level view of the PWE system and marks the boundaries of the three contributions of this study. Section III surveys the related literature and discusses the involved software and hardware components in the process. Section IV studies the integration of PWEs into the SDN paradigm and how the end-to-end system workflow is realized. Section V describes the graph-based modeling of PWEs, leading to a purely algorithmic framework for tuning their behavior, as described in Section VI. Section VII discusses open challenges and research directions while presenting novel directions toward the optimal deployment strategies of PWEs. This section also covers the topic of security, highlighting perspective PWE attack vectors and corresponding defense approaches. This article is concluded in Section VIII.

II. SYSTEM MODEL: A DAY IN THE LIFE OF A PROGRAMMABLE WIRELESS ENVIRONMENT

The PWE comprises a set of intelligent surfaces, all wired to an SDN controller for receiving EM manipulation commands via the EM API. We assume that an administrator has deployed metasurfaces in space (e.g., hangs them on walls and wires them to the controller) and then informs the controller about the position of the metasurfaces in the space. In other words, the controller knows the floor plan schematics and the location of the metasurfaces in order to operate. This assumption is made for ease of exposition and is alleviated in Section VII, i.e., once deployed, the metasurfaces can go through a calibration phase to discover each other. In other words, they can automatically learn their connectivity graph—see Section V—and not the explicit floor plan and positions, making them plug-and-playable. The SDN controller internals are detailed in Section IV, while the optimal surface deployment is also covered later in Section VII.

In addition, the PWE hosts a beacon, which is also wired to the SDN controller. The beacon can be collocated with a wireless access point, but we treat it as a different entity, for clarity. The beacon advertises the presence of the PWE to nearby devices, using an out-of-band wireless control channel to do so. This control band is not manipulated by intelligent surfaces.

We proceed to describe the general end-to-end workflow of a PWE, as illustrated in Fig. 2.

1) A device enters the intelligent environment. It listens to the control channel for broadcast messages advertising the existence of the PWE. The device can then associate itself with the PWE that it has access to using a common authorization system (e.g., as in Wi-Fi). Thus, the PWE now knows the presence of the user device in its vicinity.
   a) Optionally, upon authorization, the user device may be presented with choices regarding the service objectives it seeks (security, WPT, QoS, and so on) [9], [17], [19]. This can happen by redirecting the user device to a web page that presents those choices. This choice may be changed by the user at any time by visiting the same web interface. At this point, the control channel can be moved into the in-band, once the PWE/user device begins to operate. This can free-up resources from the PWE advertisement band while improving security and privacy.
   b) The control channel remains open throughout the service of the device by the PWE, in order to send beamforming directives to the user device, to collect performance feedback from it, to explicitly disconnect from the PWE, and so on [9], [17], [19]. For instance, the study of Liaskos *et al.* [21] proposes to deploy intelligent surfaces on ceilings and the upper parts of walls only since they constitute unused resources with easy access to a power supply. Then, a user device can always beamform upward, using the gyroscopes nowadays commonly embedded in smartphones. Time-outs can also be handled for more efficient resource management. For example, a user may have left the PWE without notification, and the corresponding metasurfaces should be freed-up. A session protocol can be used for this beacon/user device communication (e.g., a PWE variation of SIP [22]).
   c) If the user does not pass the access control check, he can be treated as an intruder or unauthorized user, depending on the setting. The reaction could be to just provide a minimum level of PWE service, treat him as a source of interference to authorized users, or even absorb his transmissions and block further access [9], [23].

In the manner described above, the SDN controller always has a view of the devices present in its PWE and their objectives. A core strength of PWEs is that they enable a graph representation of the complete environment, which abstracts the underlying physics [17]. This approach transforms the PWE configurations into a path-finding problem in a graph, which is detailed further in Section V. The exact way of how the SDN controller proceeds to serve the user objectives may be adapted to the availability of a device localization system (DLS). Two different approaches are illustrated in Fig. 3.

1) *If a DLS Does Not Exist:* The SDN controller treats the PWE configuration as a large optimization problem [24]–[28]. The variables are the cells and their states over all metasurfaces present in the area. The Tx and Rx seek to iteratively optimize the phases

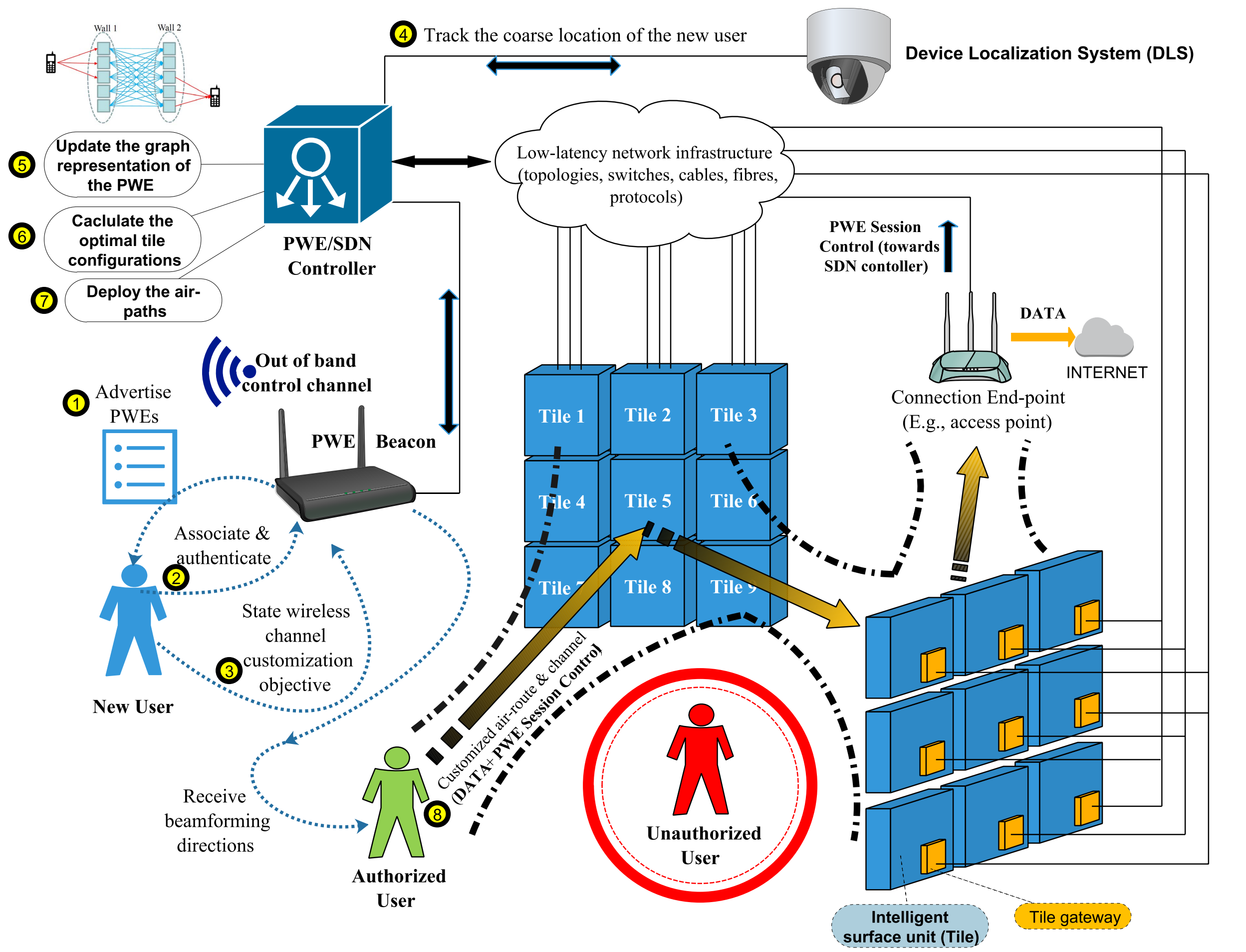


Fig. 2. *PWE end-to-end system workflow, illustrating the steps from the registration of a new user to the wireless channel customization.*

of all shifters integrated into all intelligent surfaces present in the environment [see Fig. 3 (top)]. The performance feedback received from the user devices over the control channel acts as the fitness function. A continuous optimization loop is, thus, formed, with the Rx reception quality serving as the fitness function (e.g., received power), and all phase shifter states as optimization parameters. Note that, while many excellent works treat this optimization problem [24]–[28], the real-time operation under user mobility may be inherently impossible due to the large scale of the optimization. This *flat optimization* approach is normally fit in cases where there is just one metasurface and many slow-moving users are present, and a codebook solution would not make sense to the increased complexity of the solution.

2) *If a DLS Exists:* The SDN controller also knows the approximate location of the user device. This enables the controller to perform a versatile and fast *hierarchical optimization* targeting multisurface deployments and real-time operation, as follows [2], [5], [17], [20]. Given that the surface functionalities are classified per type forming an API [see Fig. 3 (bottom)], using the graph abstraction of the PWE, the metasurfaces are first tuned to find graph paths (i.e., find coarse air-routes within the graph) that serve the general area around each user. A precalculated codebook approach, which converts the parameterized functionality types into phase shifter (or other embedded electronic elements) states in real time, is then used to pick the API callbacks and the matching initial cell states per metasurface that implements this route. Then, the same feedback/optimization loop can be established to hill-climb the codebook-derived solution. Thus, in this approach, a theoretical graph model can optimize the multitile orchestration for any type of wireless performance objective. Notice that this is a geolocation approach: the user gets associated with an area in the system, which the PWE system can know how to handle further [29].

We note the following.

1) The metasurfaces themselves can act as a very effective DLS out-of-the-box, which can accurately deduce the location of a user device, while also performing other EM functions at the same time [30]–[33].

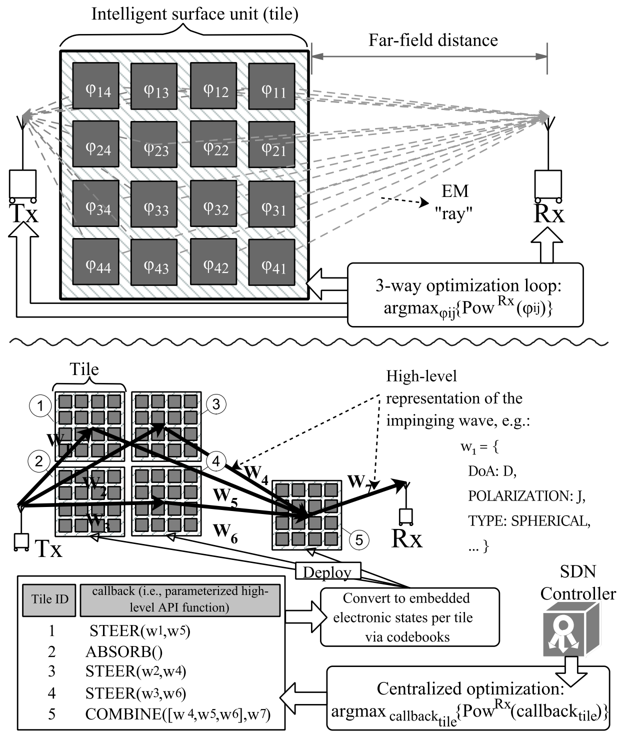


Fig. 3. ***Different PWE optimization approaches in the literature. Top: SREs favor the flat optimization approach. Bottom: PWEs favor the hierarchical approach targeting multisurface deployments and real-time operation.***

2) The hierarchical optimization approach does not replace the need to perform channel estimation. It is just an experimentally tested and highly efficient way of initiating the optimization from a promising initial solution [5]. Hill-climbing, using CSI feedback loops, takes place around the initial solution supplied by the codebook.

III. HARDWARE AND SOFTWARE: RELATED STUDIES

We proceed to present the related literature. For ease of exposition, we classify the functionality of smart radio/intelligent environments in the functionality layers, as shown in Fig. 4. At the physical layer, PWEs consider any type of metasurface equipped with a hardware gateway offering standard connectivity. At the network layer, PWEs offer the complete set of infrastructure, protocols, and workflows covering the system operation from user registration, authentication, and statement of requirements to the corresponding wireless channel optimization. The control layer is aligned with the SDN paradigm offering direct integration to existing network infrastructure. The central controller hosts software interfaces for offering an abstracted graph model of the wireless environment. Using this model, the application layer allows for customized wireless propagation-as-an-app per communicating device pair for increased communication quality, security, and wireless power transfer.

A. Physical Layer

1) Materials, Hardware and Supported EM Capabilities: Metamaterials are simple artificial structures, created by interconnected basic structures, called cells or meta-atoms [34], [35]. The planar (2-D) assemblies of meta-atoms, known as metasurfaces [36], [37], are of particular interest, given their low-cost, simplicity, and scalability in manufacturing, as, for example, using printed circuit broads, flexible materials, 3-D printing, and large-area electronics [3], [38]. Within the last decades, research and development have enabled the realization of novel components with engineered and even unnatural functionalities for the RF, THz, and optical spectra. The wide-ranging applications enabled by the engineered functionalities include EM invisibility, total radiation absorption, filtering and steering of impinging waves, and ultraefficient, miniaturized antennas for sensors and implantable communication devices. In particular, research in metasurfaces and their potential applications in communication systems has received enormous attention lately. The extensive literature is nicely laid out in a vast number of papers. A comprehensive survey is performed in [39].

The physical layer, as shown in Fig. 4, comprises the metasurface and any associated electronics to effect tunability. In the following, we provide a brief review, tracing the history from early nontunable metasurface designs right through to software-defined metasurfaces (SDMs) [40].

Early metasurfaces, due to their construction, were composed of passive building blocks, such as structured metallic or dielectric resonators [see Fig. 5(a)]. These were limited to a fixed EM response for a given specific incident EM wave (frequency, incidence direction, and polarization). Despite the inflexibility to any change in the state of the environment (e.g., incident direction of EM wave), it had proved the concept that artificial, man-made, materials are capable of manipulating EM waves beyond what nature intended. As noted in [41], many devices had been implemented, including phase shifters [27], couplers [42], beam shapers [43], [44], wave-plates [45], invisibility cloaks [46], [47], and many other functional devices. However, given the inflexibility of such materials to adapt to changing environmental conditions, dynamically tunable metasurface materials were proposed soon after, able to change the desired output functionality or to adapt the metasurface to an input wave of different characteristics [see Fig. 5(b)]. This greatly enhanced the potential for practical applications (for example, imaging, communication, and sensing) giving rise to tunable metasurfaces [39]. Historically, tunable metasurfaces evolved from globally tunable to locally tunable. In globally tunable, all unit cells are collectively controlled in the same way, as opposed to locally tunable where each constitutive

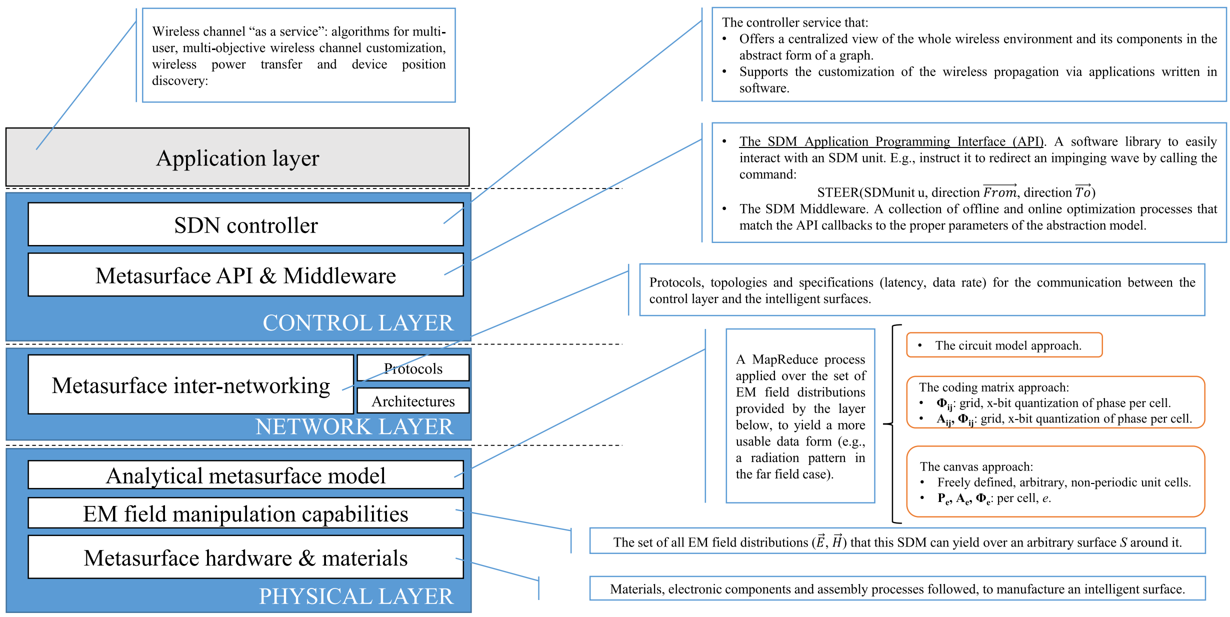


Fig. 4. ***Functionality layer approach for organizing the ongoing research efforts in the literature.***

unit cell of the metasurface is independently tuned. This allows additional functionalities to be achieved that rely on the spatial modulation of the surface impedance (e.g., wavefront control and holography) and also the ability to dynamically switch between functionalities, thus enabling reconfigurability. This later approach is naturally suited to programmatic control with a computer, which is pertinent to SDMs [40].

To achieve reconfigurability, various physical mechanisms can be used for tuning the constitutive unit cells, namely, electronic, optical, magnetic, and thermal tunings [39]. Here, we briefly discuss the evolution of locally tunable metasurfaces realized by incorporating lumped switching electronic elements in the meta-atom configuration, which with proper biasing signals can affect the surface impedance. Early designs typically used p-i-n switch diodes (offering binary or digital control) and varactors (offering analog or continuous control) to offer tunability. The usage of switching p-i-n diodes as the lumped elements allows for a binary approach to the local properties, that is, obtaining two-phase/amplitude states for the unit cell response. On the other hand, increasing the meta-atom state, for example, by using varactors, enables one to achieve a continuous variation of the reactive (imaginary) part of the equivalent surface impedance. In order to exploit the full capability of the available tuning space, both the real and imaginary parts of the surface impedance need to be made continuously tunable (see Fig. 6) [48]. This allows for complete tuning in the entire complex plane, offering independent control over both the amplitude and phase of the transmitted/reflected wave. Following this rationale, by integrating into the unit cells, a tunable chip (controller) that provides continuously

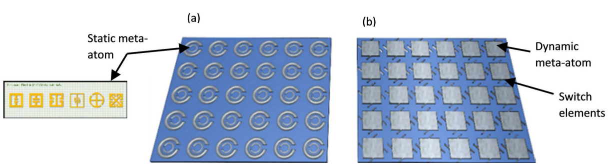


Fig. 5. ***Metasurface designs: (a) static meta-atom design and (b) dynamic meta-atom structures, featuring switch elements.***

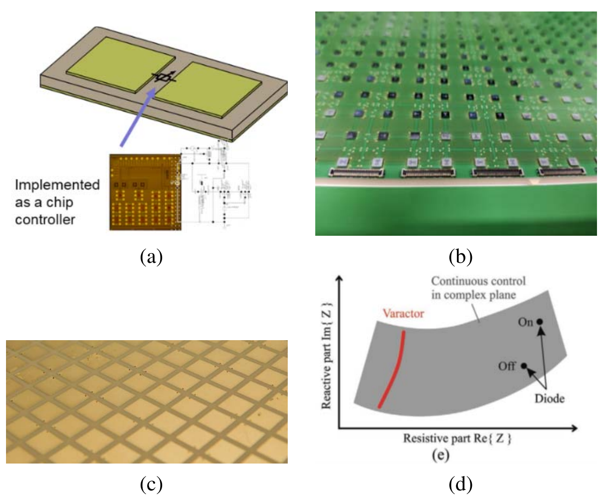


Fig. 6. ***Metasurface example: (a) example realization of controller chip, (b) fabrication of controller chips on back plane of metasurface, (c) fabrication of front plane of metasurface, and (d) operating range of tunable lumped elements: diode, varactor, and tunable controller chip.***

tunable resistance and reactance load to each of the metasurface unit cells [49], [50] allows full and programmatic flexibility, e.g., tunable anomalous reflection and tunable perfect absorption within the same metasurface, as demonstrated in [49].

As given above from the evolution of metasurfaces, the concept of a fully programmable metasurface has emerged as it appears, e.g., in software-defined materials (SDMs) [51]. To realize a fully programmable metasurface, enabling microsecond and microwatt reconfiguration of complex impedances at microwave frequencies, a CMOS application-specific integrated circuit (ASIC) with networking capabilities was developed within the Visorsurf project [40]. Its implementation details, such as the adoption of asynchronous digital control circuitry, are reported in [49]. The integration of a communication router within each tunable chip realizes an intranetwork of controllers, thus allowing software-driven programmatic control of each unit cell via a gateway connected to the external world. The SDM concept outlined above is schematically illustrated in Fig. 1. Note that third-party communication devices can also be incorporated to form an intermetasurface network to interconnect multiple metasurfaces, compatible with the emerging Internet-of-Things (IoT) paradigm. This is the approach adopted in the Visorsurf project [40], offering a system with full programmatic flexibility.

Such SDMs open the door to the realization of real-world applications and supply even more functionalities, such as synergistic actuation, sensing, adaptation, and resilience to faults (self-healing). The controllers can communicate with each other by means of a smart network of interconnected controllers. Hence, by providing a customizable arrangement of the impedance values of each individual cell, one enables the tuning of the desired collective metamaterial response of the entire tile. Harnessing these extra capabilities is a decisive step toward the massive deployment of functional metasurfaces. Furthermore, the capability of distributing and exchanging commands between the gateway and the networked controllers can be utilized to make the metasurface resilient to faults, as information can be rerouted to avoid damaged actuators or reach the intended controllers even if a set of connections fail, as analyzed in [52]. This new paradigm of programmable metasurfaces features unit cells, which, instead of mere actuators, are equipped with networked tunable chip controllers that possess actuation, communication, and sensing capabilities. For example, the capability of obtaining distributed sensory measurements, such as the current intensity over the actuators, can allow the determination of the impinging wave and, subsequently, configure the state of the actuators accordingly, so as to achieve the desirable function [30], [31].

2) Energy Efficiency: It becomes evident from the brief survey of the physical layer above that the software-defined metamaterials can be built in many ways. However, particularly, implementations based on the reflectarray concept have gained trained traction for two major reasons.

1) They can be manufactured relatively easily as PCBs.
2) The p-i-n diodes that they commonly integrate have very fast response times (≈2–3 ns [53]), which enables more advanced functionalities stemming from reconfiguring the RIS faster than the period of the impinging waves.

As a downside, reflectarrays may, indeed, yield power consumption concerns, as the p-i-n diodes need to be supplied with power when activated [7], [53]. (The same naturally applies to semiconductor-based varactors and varistors.)

Therefore, we can classify the software-defined metamaterials as state-preserving and nonstate-preserving based on the nature of the integrated switching elements. For instance, certain microelectromechanical switches (MEMSs) and microfluidic switches can provide tunable RF response while requiring energy only when changing state and not for maintaining it [54], [55]. Such elements are denoted as "state-reserving," while p-i-n diodes, varactors, and varistors are denoted as "nonstate-preserving."

These exhibit, of course, a tradeoff between energy consumption, cost, and response times. For instance, state-preserving elements offer zero consumption when unaltered but usually offer larger response times (e.g., even several microseconds [54]), which may preclude some of the advance functionalities mentioned above. [Still, such times are sufficient for updating the configuration of a tile, e.g., 20–100 times per second, i.e., likely fast enough for tracking and serving a user while moving in space (cf. Fig. 9).] Moreover, the demand for such components is certainly lower than for p-i-n diodes, meaning that their cost is likely larger in the general case.

This tradeoff constitutes a current research challenge, however, and may change drastically in the coming years. Thus, in the future, extremely energy-efficient p-i-n diodes or ultrafast and cheap MEMS may become widely available.

3) Analyzing Metasurfaces: As discussed in Section II, metasurfaces constitute the general concept of programmable physics. An impinging wave creates inductive currents over the area of a metasurface. A tunable circuit then ideally transforms this distribution into a different one, which yields a required EM response as a whole. No limitation is posed to the form of the metasurface and the circuitry. If the energy preservation principle holds, meaning that the induced surface distribution can, indeed, be transformed into the required one, then it is considered achievable.

Notice that the described process requires the backtracking of Maxwell's equations (from the required EM response back to the equivalent surface distribution that can create it). This constitutes an open challenge in physics (see [56] under Huygens metasurfaces).

Therefore, researchers have striven for the next best, tractable alternative, i.e., *circuit models* that describes the interaction between the impinging wave and the metasurfaces [57]–[59]. The premise is to model the EM interaction between neighboring meta-atoms, their surrounding materials, and the impinging wave as a set of interconnected, lumped circuit elements. The meta-atom/impinging wave is then modeled as a current source that feeds the equivalent circuit with power. The reflected waves are also modeled by lumped radiating elements (sinks) that have a degree of matching with the source currents. The benefits of the circuit model are that: 1) it captures all the major EM phenomena undergoing within and near the metasurface; 2) it can be solved with moderate computation effort; and 3) it can lead to directly interpretable insights about the operation of the metasurface. Nonetheless, a general circuit that can freely convert between surface current distributions is also hard to generalize and approach analytically. Moreover, a circuit model often needs to be calibrated with simulation or measurement-derived data.

Thus, research has turned to reflectarrays as a simplified but well-understood variant of the metasurfaces concept [60]. Reflectarrays are based on the analytical model of phase shifters. The phase shifters are independent scatterers of impinging waves, meaning that: 1) there is no control over the current flow among the shifters and 2) there is current flow control only between the shifter and the ground plane. A tunable RF impedance regulates the scattering amplitude and phase. Thus, much like a signal that can be decomposed to a Fourier series of sinusoid signals, a required departing wavefront in the far-field can also be decomposed to a corresponding set of properly shifted impinging signal variants. The implementation then follows in a straightforward fashion (commonly as large PCBs), subject to any restrictions set by modern large-area electronics. As a downside of their simplicity, reflectarrays exhibit narrowband response, subject to the standard antenna theory limitations and due to the independent nature of the shifters [61]. The degree of their control granularity over the impinging EM wave, especially regarding polarization control and whenever close to the surface (e.g., indoors), can also be limited. Increasing the density of the phase shifters may help counter the latter effect. Most importantly, the phase shifter approach is efficient for crafting an EM response that can be expressed as a scalar reduction of the EM field (e.g., power/scattering diagram or other map-reduce process outcomes in general) and not for crafting an EM vector field. Analytically, the *coding matrix approach* has been a widely successful model for formulating the configuration of a reflectarray and its resulting far-field radiation pattern [62], [63]. The coding matrix maps the phase value of each shifter into a series of bits and employs the resulting matrix as an intermediate layer of expression beyond hardware-specific attributes.

Depending on the specific implementation of a metasurface, the phase shifter approach can be extended to a *canvas* model, offering greater precision. First, the meta-atom structure itself can be tunable and able to swap between different EM scattering profiles [60]. This can provide an extra capability toward forming a required, total metasurface response with higher accuracy or with fewer meta-atoms. Second, using the same principle, the meta-atoms could be interconnectable (e.g., via MEMS) in order to dynamically form meta-atoms that resonate at a required frequency band [64].

The SDM concept remains applicable to any analytical model describing the EM capabilities of a metasurface, given that the software layer abstracts the underlying physics, offering a unified control approach, albeit with different degrees of efficiency.

B. Network Layer

The network layer comprises the communication infrastructure, protocols, and processes required for exchanging information between a set of intelligent surfaces, a central control entity, and any existing system [17]. The nature of this information, such as: 1) EM configuration commands sent from the control entity to the surfaces; 2) acknowledgment of proper operation sent to the control entity by the surfaces; or 3) user device location discovery information gathered by a third system to the control entity, is agnostic to the network layer. Instead, the goal of this layer is to provide the necessary bandwidth, latency, and error resilience to transfer this information among the involved devices. (Notably, few works have studied this layer, presented more extensively in Section III-C.)

To this end, the network layer needs to be carefully designed, especially as the number of deployed intelligent surfaces grows. First, the nature of the physical medium and the topology for connecting the gateways of the intelligent surfaces to the central control entity needs to be defined. Choices include wired or optical buses, wired point-to-point infrastructure (e.g., Ethernet) that may form a star topology centered on the control entity (for small deployments), or hierarchies incorporating buses, routers, and switches (for large deployments). Notably, in some cases, the physical medium can be wireless as well [65], assuming that it does not interfere with or obstruct the wireless channel that is being manipulated by the smart surfaces. This can be achieved, e.g., by using different wireless bands for the control and data sessions, or by carefully integrating the two sessions into unified communications' protocols, as shown in Fig. 2.

The style of the communication, i.e., the choice between multicast or point-to-point operation, is also an integral part of the network layer design. For instance, the central control-to-intelligent surfaces' communication, "downstream" direction (e.g., carrying EM reconfiguration instructions), could support multicasting, e.g., to quickly turn off tiles whose identifier is within a broadcast range, while the surfaces-to-central control, "upstream" direction (e.g., carrying acknowledgments), could be strictly point to point. Such an approach could simplify the network layer in the downstream direction since collision avoidance and medium access would be facilitated.

As it becomes evident, there exists an abundance of options for designing and optimizing the network layer. The choices per case are subject to the targeted smart environment adaptation rate, the smart surface gateway responsiveness, the scale of the deployment, and the over sensitivity to control errors (e.g., packet drops).

C. Control Layer

The control layer comprises the following.

1) A central control entity that continuously monitors and adapts the metasurface configuration to changing environmental conditions. This entity is considered to be an SDN controller due to the alignment of the control approach, as detailed later in dedicated Section IV.
2) A set of software components that allow the SDN controller to interact with the intelligent surfaces in an abstracted and unified manner [5], [51].

The aim of the software components is to make the metasurface operation easy to integrate into systems and applications. To this end, the software application component implements software libraries that enable interaction with multiple SDMs in a physics-abstracting manner. As SDMs evolve, it is expected to incorporate control and optimization techniques, heuristics, and AI techniques for determining the required settings of the actuators in order to switch between supported functionalities and execute them.

A key point in SDM is the abstraction of physics via an API that allows end-application logic to be reused in PWEs without requiring a deep understanding of physics. A software process can be initiated for any metamaterial tile supporting a unique, one-to-one correspondence between its available switch element configurations and a large number of metamaterial functionalities in support of custom-made applications.

An example of software implementation appears in [5] and [66]. It is subdivided into two integral modules: 1) an implementation of the metamaterial API that handles the communication and allocation of existing configurations and 2) the metamaterial middleware that populates the configuration DB with new data (new tiles, configurations, and functionalities). The metamaterial middleware incorporates a full GUI environment, guiding the user through a step-by-step process to produce new configurations. It utilizes all available theoretical and computational tools for the accurate characterization of a metamaterial tile. Furthermore, it offers direct access to the configuration DB, manually, via a custom-made structured query language (SQL) manager or through the automated process following a successful metamaterial characterization. Through this process, all the necessary data related to a newly produced configuration become explicitly available to the API.

It is worth noting that the addition of software mechanisms to abstract the physics enables the use of not only classical optimization techniques based on EM theoretic approaches but also other techniques, which can determine the required settings of the actuators in order to switch between supported functionalities and execute them. For example, embedded intelligence within the metasurface can model, predict, and control the operation of the metasurfaces for arbitrary functionalities and environment conditions and, hence, automatically adapt to changes in the environment without the intervention of external components. Examples include machine learning algorithms [67], [68] and classical feedback-based adaptive control for real-time control [69], [70].

D. Application Layer

Recently, we observe an explosion of studies demonstrating potential applications of programmable metasurfaces opening up new possibilities across a huge spectrum of fields, including communications, medicine, and the military. It is outside the scope of this article to present this wide spectrum of fields; rather, we focus on communications, where a paradigm shift is to make the wireless environment controllable.

As an example of such a possibility, the idea of coating objects in a setting (e.g., walls and ceilings in an office area, in advertising panels, and in building facades outdoor) with SDMs has given rise to, among others, the concept of intelligent walls [71], equipped with an active frequency selective surface, as autonomous parts of a smart indoor environment for cognitive wireless networks; surfaces coated with programmable metasurfaces to form a PWE, in which the wireless channel is shaped and optimized via software to match exactly the physical attributes and communication objectives of users [2], [19]; and the concept of reconfigurable metasurface empowered SREs, as reviewed in [72], [73], along with LISs [74], [75], intelligent reflecting surfaces (IRSs) [76]–[79], and reconfigurable intelligent surfaces (RISs) [80]. It is worth noting that several terms can be found in the literature to describe the metasurface-based coating technology with some having the same meaning and others having subtle differences (such as PWEs implying a complete system, as shown in Fig. 2), while RIS/IRS studies mostly focus on the communication gains that can be achieved by such a system.

Furthermore, several papers have appeared in the literature addressing diverse topics, such as the control and optimization of the wireless propagation environment, mitigation of path loss, multipath fading, Doppler effects, security, and experimental implementation aspects [81]–[95].

Leveraging the PWE-driven softwarization of the wireless propagation, the channel estimation process can also be expressed as an application-layer service. At their inception, PWEs sought full compatibility with the preexisting channel estimation techniques [2], [17]. In other words, even classic CSI loops were assumed to operate normally and in a PWE-agnostic manner, while the PWE altered the physical propagation characteristics of the environment. Wireless devices would then simply readapt—i.e., performs a signal quality “hill-climbing”—much like they would if the propagation characteristics of the environment had changed otherwise, e.g., due to the introduction of an object.

Since then, studies have focused on the development of RIS-aware channel estimation techniques. Briefly summarizing the survey by Danufane *et al.* [96], the RIS-aware techniques sought to decompose the (initially unified, as described above) physical propagation phenomenon into a problem of two cascaded channels: one from the Tx to the RIS, and another from the RIS to the Rx. Following this analytical approach, the following main estimation techniques have been proposed [96].

1) ON–OFF channel estimation techniques that seek to separately activate each phase shifter on an RIS unit and deduce its effect on the channel [97]–[99].
2) Alternating optimization techniques that attempt to optimizer the Tx-to-RIS and RIS-to-Rx channels in a greedy manner, i.e., by optimizing phase shifter states sequentially or in patterns, and repeating in a loop [100]. Machine learning processes have also been employed to perform the optimization [101], [102]. When the channel matrix is sparse, compressed sensing techniques can be used to speed up the channel estimation and optimization process [103].
3) Studies have also begun to study the channel estimation process under the hierarchical optimization prism, where the complete phase shifter pattern is optimized under the restriction of yielding a clear high-level functionality, such as beam steering [104].

Also worth noting that several other associated activities, too many to list in this tutorial, are appearing, such as the potential stemming from interconnected metamaterials and the perspective networking workload and control latency models [105], scaling [40], and smart environment orchestration issues [73]. Moreover, intelligent surface technology is currently studied from the aspect of integration into other existing systems. Examples include wireless power transfer [106], backscattering [65], [107], holographic MIMO [108], vehicular communications [109], EM imaging and sensing [110], [111], and even Multiphysics-as-an-app (MaaP) [112], which introduces a paradigm shift toward transforming all objects in the system space into energy flow control points, resulting in the end-to-end, software-defined multiphysics energy propagation phenomenon as a whole.

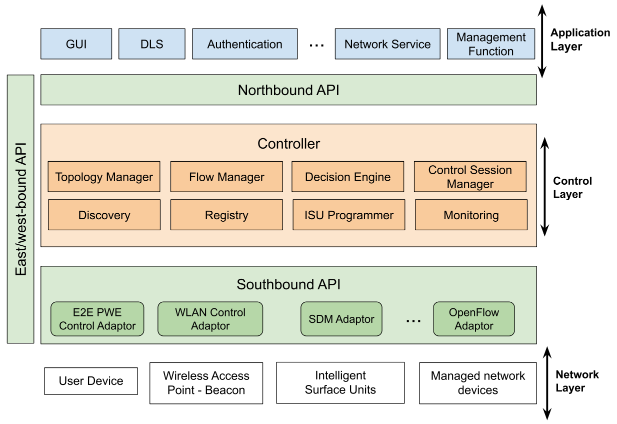


Fig. 7. ***PWE/SDN controller architecture.***

Overall, the programmable RIS concept paradigm comes as a timely step toward the introduction of the Internet-of-Materials concept, underpinning the global adaptation of IoT systems. The HSF paradigm is a step toward the Internet of Materials, extending the metasurface concept, making it accessible to a wider audience, and infusing it with novel functionality.

IV. EXTENDING THE SDN CONTROL PLANE TO PROGRAMMABLE WIRELESS ENVIRONMENTS

Here, we detail the design and implementation artifacts of a controller operating on top of an end-to-end system integrating PWEs with software-defined networks, which we call PWE/SDN controller. Its main requirements include: 1) offering “wireless-channel-as-a-service” capabilities that enable new applications or network management functionalities benefiting from the novel wireless communication performance and security features of PWEs; 2) realizing diverse performance or security objectives of multiple coexisting wireless users; 3) maintaining a global view of the end-to-end PWE system and controlling end-to-end flows over multiple hops, spanning from mobile devices and customized air-routes to SDN devices; and 4) supporting a separate control channel that realizes reliable control sessions with the mobile devices and intelligent surface units (ISUs), under ultralow delay constraints.

A presentation of the controller architecture and its core components along with an exemplary end-to-end PWE system workflow follows next.

A. PWE/SDN Controller Architecture

In Fig. 7, we provide an abstract representation of the proposed PWE/SDN controller. It adheres to the typical three-layer SDN architecture, and it supports all core controller modules, as specified in [113], so it can control efficiently SDN devices present in the network. Most importantly, its holistic operation over PWEs and SDNs is based on introducing common control functionalities for both infrastructures. For example, the controller supports a unified abstract representation of both the network topology and PWE, in consistency with the introduced theoretical framework. The alternative approach of a split architecture could have introduced additional communication and processing overhead.

Next, we describe the PWE/SDN controller interfaces and layers along with their corresponding components. The controller supports three main APIs, i.e., the *Northbound*, the *Southbound*, and the *Eastbound/Westbound APIs*. The *Northbound API* is responsible to provide "wireless-channel as a service" capabilities to network services and management functions. These include essential components for the end-to-end PWE system workflow, including the *Authentication* and *DLS* entities, realizing the user device authentication and localization, respectively. The *Northbound API* also provides the input of a graphical user interface (GUI) that visualizes the whole infrastructure and relevant performance measurements. The *Southbound API* is responsible to control heterogeneous PWE and SDN devices, including the user devices, wireless access points, ISUs, and other managed network devices. It provides a uniform API to the controller components, but its south side employs designated adapters to the particular devices, e.g., the *SDM interface* to the ISUs or an *OpenFlow API* to the SDN equipment. The *E2E PWE Control Adapter* is responsible to handle the out-of-band control communication of the controller with the devices. The controller also supports an *Eastbound/Westbound API* that enables its distributed deployment, for improved scalability, fault tolerance, and performance, like many well-known SDN controllers, e.g., ODL or ONOS [113]. For example, a controller node may be deployed near each room or corridor, so the communication delay with the ISUs is minimized.

In between the *application layer*, i.e., covering the network services and management functions benefiting from our infrastructure, and the *infrastructure layer*, i.e., being associated with the heterogeneous devices supported by the PWE/SDN controller, resides the *control layer*. The latter supports the following components.

1) *Topology manager* maintains an abstract representation of the network environment, including the PWE and the low-latency network infrastructure for both the control and data channel. It is also responsible to trigger topology discovery processes or request decisions/actions relevant to topology management.
2) *Flow manager* handles end-to-end flow control that involves all devices supported by the *infrastructure layer*, i.e., user devices, wireless access points, SDN equipment, and the PWE.
3) *Decision engine* is responsible for taking all controller decisions regarding end-to-end communication, including PWEs, as well as orchestrating the internal controller components. For example, it translates the service objectives originating from the user devices into particular optimal ISU configurations. It is a policy-based component that also supports optimization mechanisms.
4) *Control session manager* handles concurrent control sessions for multiple user devices based on a PWE variation of the SIP protocol, including the detection of inactive user devices using appropriate time-out-based mechanisms.
5) *Discovery* is a generalization of the typically used link discovery component in SDN controllers to support also dynamic ISU and mobile user device discovery. Although users initiate a system registration process themselves, the *discovery* component may be used to detect other present devices or even potential intruders.
6) *Registry* keeps track of every active user device and other infrastructure, including their configuration and expressed requirements. This information is complementary to the relevant status information maintained by the *topology manager*.
7) *ISU programmer* is responsible to handle the ISU programming via the appropriate interface, i.e., the *SDM API*.
8) *Monitoring* handles all required monitoring processes, including the adherence of the system to expressed service performance goals from the users.

The message exchange sequence diagram of Fig. 8 illustrates how the proposed PWE/SDN controller realizes the end-to-end PWE system workflow of Fig 2. In a nutshell, the workflow involves three main steps: 1) user authenticates and associates himself/herself with the PWE system; 2) the system realizes the expressed service performance or security objectives from the user; and 3) it establishes appropriate data communication, control session, and monitoring processes.

In more detail, such steps involve a number of PWE/SDN controller components, supporting network services, or management functions. User authenticates with the assistance of the *WAP/beacon* and *authentication*, which are external to the controller entities. The association of the user device with the system is being realized by the *registry* controller component. Next, the user expresses its wireless channel customization objective to the *decision engine* controller component via the *WAP/beacon*. The former component triggers the localization of the user device through the external DLS entity, which responds with the coarse device location. Now, the *decision engine* is ready to update the PWE graph representation handled by the *topology manager* and take appropriate optimization decisions regarding the required ISU configurations. Such configurations are the input of deployed air paths via the *ISU programmer* component. The workflow completes with the realization of a control session and a monitoring process through the *control session manager* and *monitoring* components, respectively.

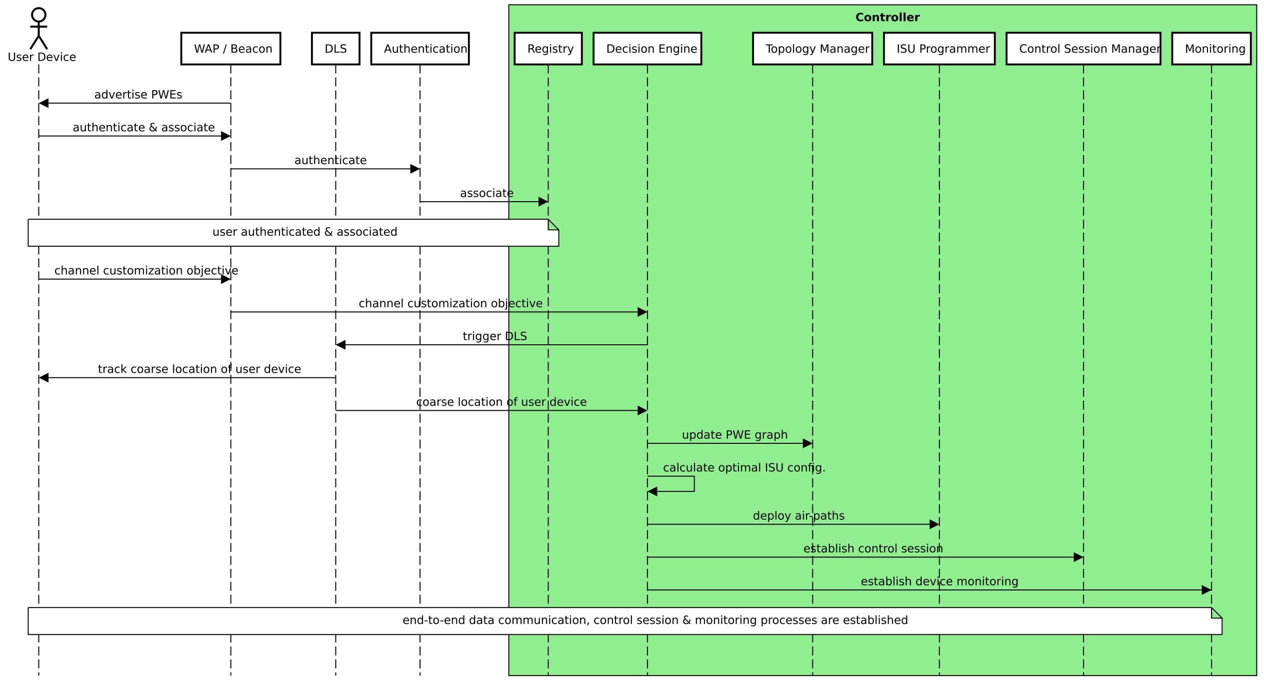


Fig. 8. ***Message exchange diagram of exemplary end-to-end PWE system workflow.***

We now provide an initial investigation of control channel aspects.

B. Control Channel Design Issues

Here, we provide basic details on the control channel characteristics, its main requirements, and relevant implementation issues.

The PWE/SDN controller communicates via a separate control channel with: 1) all intelligence surface units in order to execute particular EM functions and 2) mobile devices being present in the area, e.g., to associate users with the system, pass beamforming directives, and collect performance feedback from them. Consequently, there is a need for fixed or wireless communication with the ISU grid and for wireless communication with the mobile devices in the area. The wireless control communication uses a separate radio band that is not manipulated by the intelligent surfaces and can be moved into the in-band once the PWE/user device begins to operate. This can free-up resources from the PWE advertisement band and improve security/privacy.

The ISUs form a grid topology and are being controlled by one IoT device embedded in each one of them. The devices communicate and form a closed, isolated system with the controller, which should be secure, since the proposed system can be used in security scenarios. Although the topology discovery process of the IoT devices may be straightforward, the controller should also be able to locate their physical positioning and which aspect could be handled either manually or via a dynamic positioning system. The latter process is considered a future work.

The control channel requirements may match the characteristics of the particular PWE environment. We assume three categories of ISU grid installations: 1) semipermanent (e.g., for an exhibition hall) or hanged on regular walls without major refurbishments; 2) on a smart-home or building; and 3) on a highly critical environment. The above cases have distinct requirements, which we elaborate on in the following.

In the first deployment case, a wireless control interface improves the flexibility of installation, i.e., avoids wall refurbishments to accommodate control channel cabling. This approach reduces the cabling costs and potentially the complexity of the involved controller because the same communication technology could be applied for both ISUs and mobile users.

Regarding the case of smart-homes or buildings, many of them are equipped with fake walls, as well as with centralized control buses for all appliances. The latter could accommodate ISUs. For example, the IMEC HomeLab testbed [114] employs the relevant VelBus and OpenHab technologies.

Furthermore, a highly critical environment is characterized by strict security requirements, so a wired control channel connectivity is the main option. In these environments, physical access to cabling is prohibited, e.g., the cabling may be built inside the wall. Potential cable faults can be mitigated with redundant connectivity, as typically used in data center networks.

The number of deployed ISUs can be high since the areas to be covered may be large. The large-scale and dense deployment of ISUs may cause interference issues

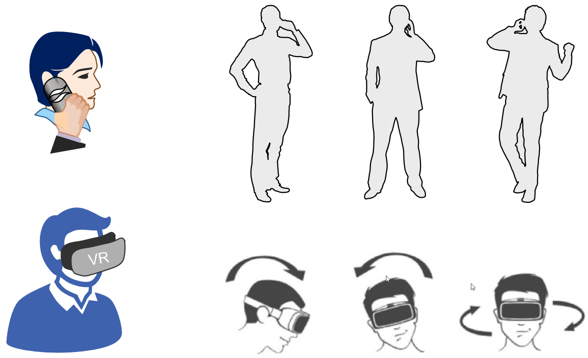


Fig. 9. ***Parallelism between cell-phone and AR/VR user mobility. The similarity in the user motion may imply similar environment control latency requirements.***

and collisions that should be handled from the relevant communication facilities. Furthermore, an ultralow delay communication requirement is essential.

We suggest that the end-to-end delay of the control channel should be lower than 20 ms, i.e., the delay threshold is expected to be similar to the one defined for the AR/VR concept, since the latter and PWEs have inherent similarities in user/wireless device mobility [115], as shown in Fig. 9.

The propagation delay in a specific LAN with a given diameter is low and deterministic (i.e., in the range of nanoseconds). Furthermore, the processing and transmission delays in commodity Ethernet switches are in the range of microseconds or below. Consequently, the most important part of end-to-end delay that can be improved is the nondeterministic queuing delay [116].

Along these lines, we now investigate the delay requirements of the discussed SDN/PWE system workflow. The initial and periodic communications between the user and the controller seem to be mostly associated with relaxed delay constraints, e.g., for the association with the system, the communication of user requirements, and the periodic reporting of measured QoS or QoE. Ultralow delay demands characterize the processes that decide and realize the custom air paths. For the same reason, we mainly focus here on the delay constraints of the controller-ISUs' communication. Furthermore, the wireless communication with the user device also involves delay-sensitive aspects of the air path customization, e.g., the mobile user positioning, the communication of beamforming directives, and the networking workload and control latency [105], and scaling [40]. We consider such aspects as future work.

Consequently, there is a need for a dynamic resource reservation process associated with the control channel session. The reservation should be triggered with the association of a new user with the PWE system and the release of resources when the user leaves, e.g., after an explicit disconnection or a time-out event. There is a need to allocate an appropriate amount of network, i.e., for the ultralow delay communication of control messages, and processing resources, i.e., to enable rapid decision-making. The network or edge cloud establishment (i.e., hosting the PWE/SDN controller) may be dimensioned according to the characteristics of the area to be covered, e.g., size, average, or maximum people occupancy.

As a bottom line, the PWE/SDN control channel should be scalable, reliable, and secure when demanded to support ultralow delay communication through dynamic resource reservation.

Due to the resemblance of the above requirements with those originating from industrial systems, we investigate relevant communication solutions. In this context, we see three main categories of candidate communication technologies for the PWE/SDN control channel: 1) fieldbus and real-time Ethernet protocols; 2) time-sensitive networking (TSN) and deterministic networking (Det-Net) approaches; and 3) real-time SDN proposals. These technologies are briefly elaborated on in the following.

Fieldbus serial communication protocols are being used in the industry for many years, such as control area network (CAN), Modbus, and PROFIBUS. For example, CAN is used in automotive systems and as a central bus in smart-home deployments [114]. However, these widely used approaches are characterized by bandwidth, physical range, and addressing space limitations [117]. The real-time Ethernet protocol is a natural evolution of Fieldbus approaches, with standardized, low-cost solutions, as well as higher flexibility, communication throughput, improved distance, and support of more nodes. Example protocols are EtherCAT, PROFINET, Sercos, and FTT-Ethernet [118].

Fieldbus or real-time Ethernet technologies may not be adequate to provide the level of integration, large-scale operation, and real-time flexibility demanded by PWE systems, but they can be used in particular small-scale PWE deployment cases. However, improving these aspects requires deploying distributed control systems or gateways that increase cost, latency, and management complexity. There is a tradeoff between real-time performance and run-time flexibility [117], which should be considered.

A recent approach enabling real-time communication in relevant systems to PWEs is TSN [119], [120]. TSN brings bounded latency, improved reliability and zero congestion loss through bandwidth, buffering, and scheduling resource reservations for particular traffic flows. It also implements time synchronization among devices and contracts between transmitters and the network. Furthermore, TSN offers centralized management of relevant devices. For example, it exports a YANG/NETCONF interface that enables node topology/capabilities discovery and configuration of TSN features, whose aspect allows the incorporation with a centralized PWE controller.

There is a number of recent proposals extending a TSN domain with wireless nodes, e.g., [121] and [122], which may be used for the communication of PWE/SDN controller with mobile users. Wi-Fi 6/6E and URLLC 5G capabilities could be employed for such communication,

Table 1 Comparison of SDN Controllers Under Consideration

	Infrastructures considered	Architecture	Modularity	Low latency	Mobility	Out-of-band ctrl
ODL [131], [132], [133]	Fixed, IoT / WSN, VANET	Distributed Flat	High	X	X	
POX [134], [135]	Fixed, IoT / WSN	Centralized	Medium			
Floodlight [136], [130]	Fixed, TSN	Centralized	Medium	X		
ONOS [137], [138]	Fixed, IoT / WSN	Distributed Flat	High			
SDN-WISE [139], [140]	IoT / WSN	Centralized	Medium	X		
Atomic SDN [141]	IoT / WSN	Centralized	Medium	X		
Soft-WSN [142]	IoT / WSN	Centralized	Medium			
VERO-SDN [143]	IoT / WSN	Distr. Control Gateways	Medium			X
SD-MIoT [144]	IoT / WSN	Distr. Control Gateways	Medium		X	X

as well as time-synchronized channel hopping (TCSH) in the case of IoT-based operation.

Currently, the IEEE 802.1 TSN Task Group [123] has implemented relevant significant standardization work, while the IETF Det-Net Working Group [124] focuses on layer 3 aspects of real-time communication. However, these solutions do not scale well yet. Relevant issues on TSN deployments are documented at [125] and are also being improved in the context of Det-Net.

In our understanding, existing initiatives to support real-time communication in SDN architectures consider the integration of real-time Ethernet or TSN solutions with SDNs. Indicatively, FTT-OpenFlow [126] applies FTT-Ethernet principles to OpenFlow and improves the performance of sporadic real-time traffic. OpenFlow-RT [127] also considers the FTT paradigm and enables the specification of real-time traffic flows and reservations. PROFINET integration is introduced in SDPROFINET [128]. Furthermore, the works presented in [116], [129], and [130] target integrating TSNs with SDNs, e.g., Boehm *et al.* [130] introduce an interesting TSN/SDN architecture that seems well-aligned to the main requirements of the investigated PWE/SDN system.

In summary, the fieldbus or real-time Ethernet solutions could be applied in small-scale PWE environments, especially in smart-homes or buildings that already support relevant capabilities. TSN is a promising technology that offers the needed real-time communication flexibility and reliability; however, it faces scalability issues. Scalability and flexibility are improved with the real-time SDN solutions; however, they are not as complete as TSN in terms of real-time performance capabilities. SDN brings also another advantage. Their inherent multicasting capability could be utilized in PWE environments, e.g., the tiles for a particular user, or a specific wall or room could be assigned a single multicast address.

C. Controller Implementation Considerations

The PWE/SDN controller could be potentially implemented as an extension of an existing controller. Thorough surveys comparing and detailing the characteristics of the most important SDN controllers can be found in [113] and [145]. Here, we provide an initial investigation of the available options offering a scalable operation of a PWE/SDN controller while being extendable to support features that are the same or similar to those required to implement the missing PWE control aspects.

In Table 1, we enlist important proposals that are multithreaded and extendable, and have been shown to support challenging communication environments, including resource-constraint IoT or wireless sensor networks, time-sensitive networking deployments, and vehicular networks (i.e., VANETs). This last aspect demonstrates implicitly their capability to support the investigated paradigm. The table categorizes the SDN controllers according to: 1) which types of infrastructures are supported by the controller; 2) the adopted architecture type, where distributed approaches can allow deployment at larger scales; 3) their modularity level, characterizing their extendability; 4) their focus on addressing low latency requirements; and 5) whether they consider node mobility and a separate control channel. The above characteristics may be part of the basic designs/codebases or extensions of it. We provide references in support of the documented controller features.

The first four SDN controllers of Table 1 have been designed and implemented for fixed OpenFlow networks; mainly, however, the relevant architectures either initially support (e.g., ODL and ONOS) or can be extended with (e.g., floodlight) additional south interfaces. For example, Al-Rubaye *et al.* [132] introduced the support of a smart-grid IoT environment to ODL and [135] brought to POX the capability for a rapid introduction and deployment of new IoT services based on reusing various resources, e.g., devices, data, and software. The remaining solutions are focusing on IoT deployments and support OpenFlow-inspired protocols that consider the requirements of the particular environments.

Among the investigated solutions, ODL and ONOS adopt a distributed flat architecture, bringing scalability, and resiliency advantages. Although VERO-SDN and SD-MIoT controllers are centralized, they employ distributed control gateways in order to support large networks. In this context, the location of the controller replica/node or control gateway is important for the controller performance, so it is an aspect that requires optimization.

The modularity level is high for the heavy-featured ODL and ONOS, and medium for the other controllers. Although it may be easier to extend a modular solution with new functionalities, e.g., SDN-WISE has been integrated into ONOS [138], however, the number of supported components may introduce additional performance overhead. For example, Zhu *et al.* [113] document a lower average RTT (i.e., communication delay between the controller and switches) in the case of more simplistic controllers (e.g., POX and Floodlight). Consequently, extending a simple implementation with essential PWE control capabilities is also an attractive option. For example, paper [130] extended floodlight to support TSN features.

Chekired *et al.* [133] introduced ultralow delay support in ODL, although focusing on VANET environments. Orozco-Santos *et al.* [140] augmented SDN-WISE with low-delay communication features based on using the TCSH channel access method and slicing for flows with different QoS requirements. Atomic SDN [141] is another solution for low-latency control in low-power wireless networks, focusing on applications with highly robust communication demands and unpredictable traffic patterns. Additional low-latency considerations for SDN controllers are documented in [146], highlighting that many controller aspects could be improved that impact on delay, including load balancing, congestion control, traffic, and flow table management. Another important issue is to apply intelligent prediction of network conditions, so proactive strategies can be deployed, i.e., reducing the time-consuming involvement of the controller.

Besides paper [133] that considers mobility issues due to its focus on VANET networks, work [144] introduced mobility handling mechanisms, including for topology discovery, topology control, and mobility prediction of IoT nodes. Such features could be the basis for the controller communication with the mobile user devices, as well as for handling the relevant data plane communication. Furthermore, papers [143], [144] use out-of-band SDN control of IoT devices and demonstrate particular advantages. These insights could be useful for the out-of-band control channel considerations of the envisioned controller.

As a bottom line, in the case of a wireless control channel, a potential choice could be to implement SD-MIoT [144] features in the ONOS controller, in a similar way that SDN-WISE [138] was integrated, while also using TCSH to enable low-delay communication. Regarding the fixed network control channel option, the support of TSN capabilities in floodlight [130] is a good starting point.

V. PHYSICS ABSTRACTED: A THEORETICAL MODEL FOR UNDERSTANDING AND MANIPULATING PWEs PURELY ALGORITHMICALLY

This section discusses the theoretical and algorithmic challenges underlying the central control and optimization of PWEs. First, we detail a graph-based model that captures PWE behaviors, transforming communications between users to paths on the graph. This abstraction enables the application of classic graph algorithms to optimize PWEs, for example, providing the shortest tile routes between users. Indeed, an attractive feature of PWEs is their ability to adapt to changes, e.g., to support user movement, join, and leave, as well as policy updates. However, the update of PWE configurations also introduces novel challenges. In particular, updates may occur *asynchronously*, even if communicated simultaneously from a controller, due to communication and reconfiguration delays. Given the strict performance and security requirements of PWEs, it is critical that these updates are scheduled to avoid undesirable transient states, such as loops or communication leaks. As we will show, the problem is related to the consistent network update problem in software-defined networks, which has recently received much attention [147]; however, it comes with several specific and different constraints in the PWE context. We will illustrate and formalize these challenges and discuss algorithms and complexity results.

A. Graph Abstraction

We model a PWE with a weighted directed graph $G = (V, E, d, g)$. In our model, the set V denotes the nodes of G representing both users and tiles; the set E denotes the edges of our graph representing communication links within the set of tiles and between users and tiles. We define d as the weight function of an edge, considered as transmission delay between two endpoints of the edge. We also define a weight function g for a tile node representing the percentage of power that it could reflect. In more detail, the components of our graph representation are as follows.

1) HyperSurface Tiles: Our focus is on multisurface environments consisting of fixed HyperSurface tiles in a 3-D space. Formally, consider the set $H \subset V$ representing HyperSurface tiles deployed in PWE. We assume that the positions of all tiles are fixed. Any tile $h \in H$ has a gain $g(h)$ representing the percentage of power it reflects from an incoming wave. The difference between gains of tiles is due to variation in their coating [38].

2) User Devices: We consider mobile user devices $U \subset V$ in the same space as tiles. Unlike tiles, user devices can change their position. Each user device is either a member of transmitter nodes $T \subset U$ or a member of receiver nodes $R \subset U$. The union of user devices and HyperSurface tiles create the nodes of our graph.

3) Connections: The connection between tiles and between tiles and users is determined based on the functionalities of tiles. We focus on the functionalities that affect *direction* of EM waves transferred in our system.

Each tile has a *steering* function to change the direction of an incoming wave. Steering functionality can be seen as virtually rotating a tile to change its reflection angle. A tile with *combine* functionality assigns a single output direction for multiple waves. Also, a tile can *absorb* an incoming wave. A tile with absorption functionality behaves similar to a blocking object, which is an important functionality for interference minimization.

Formally, a tile $h_1 \in H$ is *connectable* to another tile $h_2 \in H$ if there is no obstacle between h_1 and h_2, and h_1 can redirect an impinging wave to h_2 such that h_2 receives a significant portion of the power of the impinging wave. Similarly, we define the connectability of a tile to a receiver. A transmitter is *connected* to a tile h if h does not fully absorb the wave or h does not appear in the line-of-sight of the transmitter.

We include both tile-tile/tile-receiver connectability and connected transmitter tiles as an edge $e \in E$. We consider transmitter-tile edges directed toward the tiles, tile-receiver edges are directed toward receivers, and tile-tile edges are bidirectional; however, we assign a direction to them afterward. We note that some of the current technologies allow for a bidirectional connection between tiles as well [148]; however, in our model, we focus on the directed scenario for generality. In the pair i, the receiver node r_i connects to transmitter node t_i via set of *node disjoint* paths, denoted by P_i.

4) Weights: The weight of a tile node $h \in H$ is defined as the power gain $g(h)$ of that node. The power gain of a tile is determined based on its coating. For an edge $e \in E$, we assign a delay $d(e)$ function as its weight. The delay of each edge can be proportional to the distance between its two endpoints, or it can be adjusted based on the policies. We compute that delay of a path as the sum of its edges, and the power gain of a path is the multiplication of the power gain of its tile nodes.

5) Deployment: Consider an assignment of functions to tiles. We call the tile nodes that participate in the communication between a transmitter–receiver pair as *active nodes*. As the description of the tile uniquely determines the direction of the outgoing edges, we name the chosen edge as an *active edge*. Hence, there is a one-to-one correspondence between the tile functionalities and the set of active edges. However, we note that, after assigning functionalities to active tiles, there might be inactive tiles that accidentally receive waves. We optimize for such scenarios by running our algorithms again, considering only such inactive tiles as the set of available tiles.

6) Performance Objectives: Based on the user requirements, each transmitter–receiver pair informs the PWE about its set of objectives to be deployed. In the following, we present some fundamental performance objectives. We see these as examples, and depending on the scenario, additional objectives may be relevant [149].

a) Power transfer maximization: The first objective aims to maximize the amount of power transferred. To this end, for an incoming path to a receiver i, we multiply the gain of all tile nodes in that path to obtain the percentage of power, which can be received by the receiver i from that path. The goal is to find the set of disjoint paths that maximize the power received by a receiver. This objective is attractive, especially for wireless power transfer [19]. We define a variable ϕ_i as the percentage of the initial power received by a receiver i and define Φ as the union of power transfer objectives.

b) Quality-of-service optimization: Among interpretations of optimizing Quality of Service (QoS), we consider the useful signal over the interference ratio. Useful signals are only received through paths with bounded delay. Hence, we divide the power transferred to the receiver i into two parts: Φ_i^{useful} as the useful power from paths with bounded delay and $\Phi_i^{\text{interference}}$ from the paths that cause interference. The goal is to maximize the ratio:

$$\frac{\Phi_i^{\text{useful}}}{\Phi_i^{\text{interference}}}.$$

c) RMS delay spread minimization: In wireless communication, in order to minimize multiple path fading, it is common to minimize *root-min-square (rms)* (standard deviation) of delay spread weighted by the power transferred by the waves [150]. In our PWEs' model, assuming all the tiles have similar power loss, the same effect could be achieved.

B. Update Abstraction

PWEs do not only enable optimization for a *given* setting but also support *dynamic adjustments*. This is attractive, as adaptions are relevant in many scenarios. Few examples are given in the following [147].

Updates in paths between transmitters and receivers are, hence, inevitable. However, realizing such adaptions is nontrivial, as both the communication and the implementation of updates can incur delays. If not scheduled carefully, this can, in turn, lead to transient inconsistencies, harming performance, and security. For example, reconfigurations should be scheduled such that it is ensured that eavesdropping is avoided *even during the update*. Further critical invariants to be maintained during the update include loop freedom and congestion freedom, which can lead to significant throughput degradation and packet loss if ignored. Given these constraints, the update schedule should be optimized, for example, in terms of the time that it takes to update the network (the shorter, the better) or the number of reconfigurations that are required (the less, the better).

In general, the problem of adapting PWEs is related to the consistent SDN update problem [147]: the problem of how to schedule updates to SDN switches to consistently change one or multiple routes. However, there are several critical differences between SDNs and PWEs that significantly change the underlying algorithmic problem. In particular, while SDN switches typically come with powerful match-action tables or even processing capabilities [151]–[153], tiles are simple forwarding devices. In particular, tiles do not allow to distinguish between different flows or account for header information or tags, hence ruling out most existing update approaches [154]. Not only the capabilities of the two scenarios differ but also the requirements: the fact that tiles are located in space, communicate in a wireless fashion, and introduce additional concerns, such as eavesdropping. That said, as we will see in the following, some results from the SDN literature are still relevant here.

1) Update Objectives: PWEs, such as other networking environments, demand fast updates at a low cost. We define these two natural update objectives as follows: minimizing update cost is related to the number of tiles that are going to be updated, and update duration, which is based on the number of rounds (in which group of tiles can be updated together). We note that the following discussion also serves as an example, and several variants of these objectives may be relevant in practice to support the operator requirements.

a) Minimizing update cost: Changing tiles multiple times during an update can be undesirable not only because of the introduced delay but also because of the way such changes affect the flow forwarding. We, hence, consider schedules that only update tiles once during a schedule and aim to minimize the number of tiles that we need to update to realize a change. We also note that the objective of minimizing the number of updates in network devices is also relevant in the context of SDNs [155]; however, the difference in our model is that an update in one tile changes the outgoing edge for all incoming paths.

b) Minimizing update duration: A critical metric in the context of PWE update problems is the duration of the update schedule, i.e., we require a minimal number of interactions with the controller. We note that the two-phase update method used by Reitblatt *et al.* [154] in the context of SDNs is not applicable here as their approach requires packet tagging, which is not available in PWEs. Also, other approaches used in the SDN context [147] are different since, in SDNs, each switch can be part of multiple paths without any restriction.

One may distinguish between two natural models. In the first one, the new paths are given a priory, and in the second model, the new paths may also be subject to optimization (given certain constraints). In the following, we will focus on the second model since it is more general.

2) Update Constraints: During the PWE updates, we want to ensure that all the waves sent by transmitters can be received at their destination. Hence, we define the following constraints for any update schedule.

a) Loop freedom: During the update schedule, individual tile updates may happen at different times, which can cause transient *forwarding loops*. In order to avoid loops, we divide tile updates into *update rounds* such that tiles in one round can be updated together, but tiles from different rounds should be updated sequentially. You can see an example of when two rounds are required in Fig. 10.

b) Scatter freedom: In order to ensure that all the waves sent by transmitters reach at least a receiver, we want to avoid the cases in that the wave reflects at a tile going nowhere, i.e., "scatters" without reaching a receiver. We refer to this property as *scatter freedom*, which is similar to the notion of *blackhole freedom* in SDNs: the requirement that there always exists a forwarding rule [156]. In SDNs, this requirement alone could easily be solved by defining a default rule, and similarly, there is a simple solution also for PWEs: scatter freedom can be achieved if we assume that all tiles have absorbing functionality by default, and we only assign steering functionalities to a tile if it is part of the path from transmitter to receiver, and transmitters only send waves to tiles that can be part of a path to a receiver. However, designing a fast loop-free and scatter-free update schedule is challenging and requires extra attention.

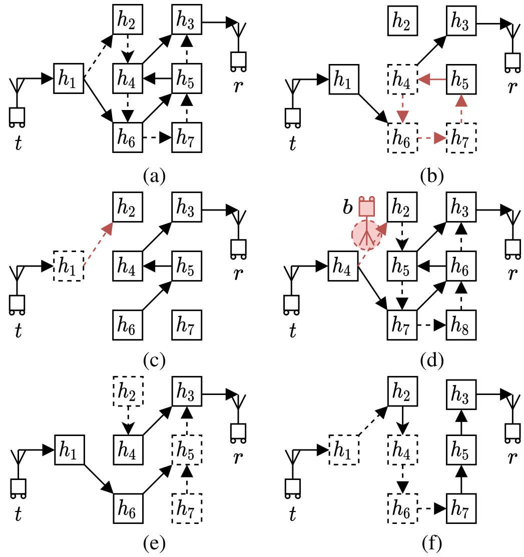


Fig. 10. ***(a) Example of the violation of update constraints. The solid lines show old paths (and the dashed ones are updated paths) between the transmitter $t \in T$ and the receiver $r \in R$. Each tile is represented by a square and each updated tile by a dashed square. (b) Updating tiles h_4, h_6, and h_7 in same round creates a transient loop. (c) Updating tiles h_1 before or in the same round as tile h_2 may cause the wave to reflect in an unwanted direction; hence, it violates scatter freedom. (d) Consider an unauthorized user $b \in B$. In the presence of user b we need to choose an alternative updated path, excluding the connection between h_1 and h_2. (e) and (f) Example of a consistent update in two rounds such that, regardless of order of tile updates in a round, the receiver remains connected to the transmitter.***

c) Eavesdropping prevention: To avoid unauthorized users from intercepting the communication between a designated receiver, we define an eavesdropping prevention objective. Formally, for each receiver, i can denote a set B_i of users that are unauthorized to interfere with communication to receiver i, and for each of such users $b_j \in B_i$, we restrict a 3-D space around it. We denote this area by $S(b_j)$. Therefore, we want paths of user i not to intersect with any of the areas surrounding unauthorized users. For an example, see Fig. 10.

Algorithm 1 Update Optimization

1: **Minimize** Δ or Λ or Φ

2: $\sum_{h\in H}\sum_{\delta\le\Delta} x_h^\delta = \Lambda$
{ Limiting the total number of tile updates to Λ }

3: $\sum_{\delta\le\Delta} x_h^\delta \le 1 \qquad \forall h\in H$
{ Each tile must be updated at most once }

4: $a_{(t,v)} = 1 \qquad \forall t\in T, (t,v)\in E$
{Outgoing edge of all transmitters must be active}

5: **for all** $\delta\le\Delta$ **do**

6: $a^\delta_{(v,w)} \ge \sum_{(u,v)\in E} a^\delta_{(u,v)} - \sum_{\delta'\le\delta} x_v^{\delta'} \qquad \forall (v,w)\in P^o$
{Setting activity of the edges in old paths}

7: $a^\delta_{(v,w)} \ge \sum_{(u,v)\in E} a^\delta_{(u,v)} + \sum_{\delta'\le\delta} x_v^{\delta'} - 1 \forall (v,w)\notin P^o$
{Setting activity of the edges that are not in old paths}

8: $a^\delta_{(v,w)} + \frac{(o_v-o_w)}{|V|-1} < 1 \qquad \forall (v,w)\in E$
{Enforcing loop-freedom constraint}

9: $a^\delta_{(u,v)} \le \sum_{(v,w)\in E} a^\delta_{(v,w)}$
{Enforcing scatter-freedom}

10: $\sum_i \prod_{h\in P_i^r} g(h) \ge \phi_i \qquad \forall P_i^r \in P^r$
{Selecting paths with sufficient percentage of power}

11: $a^\delta_{(v,w)} \cap S(b_j) = \emptyset \qquad \forall b_j\in B_i, P_i\in P, \delta\le\Delta$
{Block eavesdropping}

VI. ALGORITHMS

We now discuss algorithms to optimize PWEs in the different models introduced above. We first present an optimal solution approach, based on a mixed-integer program, and then discuss fast algorithms.

Before going into detail, we like to point out that the following algorithms aim to combine techniques for solving two classical problems, namely, dynamic shortest paths and consistent network updates. Variations of dynamic shortest path problems have been studied for more than half a century [157], [158], while consistency in network update has received a lot of attention recently with the emergence of software-defined networks [147], [159]. We hope our first steps in this combination pave the way for further research in optimizing PWEs.

A. Mathematical Programming Algorithms

We can formulate a *mixed integer program* that either optimizes the number of tiles that are updated (denoted by Λ) or the number of rounds that are required (denoted by Δ); combinations are also possible. Our algorithm requires a few variables for each node and edge of the graph.

1) Each tile node h has a Boolean *tile update variable* x_h^δ, which equals to one if tile node h gets updated in the round δ.
   a) We assign an integer o_h^δ in range $[0,|V|-1]$ as the *order* of each tile node h.
   b) For an edge (v,w), we define Boolean *activity variable* $a^\delta_{(v,w)}$, which is equal to one if node (v,w) is active in round δ.

We consider a set of initial paths as old paths, P^o, such that the user's objectives and constraints are respected in the initial set of paths.

Before running the program, we need to fix the values of two of three objectives. We also assume that the old paths P^o are given. Constraint 2 ensures that the number of updated tiles is limited by Λ, and Constraint 3 ensures that each tile is updated at most once. We then set all the outgoing edges of a transmitter to be active (see Constraint 4).

If edge (v,w) is part of the old paths, there exists an active edge to its tail node v, and it is not yet updated, this edge would remain active (see Constraint 6). If it was not already part of the old path, it could only become active if its tail updates, and there is also an active edge to v (see Constraint 7).

To guarantee loop freedom, in any active edge (v,w), the order of its tail v must always be lower than its head w. Therefore, when edge (v,w) is active, then we need to have $(o_v - o_w) < 0$. As value of the node orders is in the range of $[0,|V|-1]$, if $a^\delta_{(v,w)} = 1$, then we have $-1 \le \frac{(o_v-o_w)}{|V|-1} < 0$. Hence, in a loop free update schedule, we always have Constraint 8.

Constraint 9 ensures that the incoming wave impinging from an active reaches another tile or a transmitter. To maintain the power transfer objectives, we check the sum of power that arrives at a receiver to the threshold set at the beginning of the integer linear program (LP) (see Constraint 10).

For eavesdropping mitigation, we need to avoid edges that pass through the area surrounding unauthorized users (see Constraint 11).

In the end, the set of active edges in the last round creates the updated paths that we aimed for.

B. Fast Algorithms

Our mixed-integer program provides an optimal solution; however, its running time can be impractical for large PWEs. Today, only relatively little is known about fast algorithms that provide exact solutions. However, there may exist solutions that can be found quickly and which are not far from optimal.

1) Relaxation and Rounding: Given our integer programming formulations, a natural approach is to use linear programming relaxation (LPs that can be solved in polynomial time) and then round those solutions back to integer efficiently (e.g., using randomized rounding).

2) Greedy Optimization: Our static objectives are related to finding the shortest paths. However, as each tile node can participate in a limited number of paths, we cannot deploy the shortest path of all transmitter–receiver pairs at once. One solution is to consider an ordering of transmitter–receiver pairs based on their distance and then prioritize the ones that are further from each other [17], but ordering is also possible based on other measures. We iterate over the transmitter–receiver pairs based on their order and then apply one of the well-known *k-shortest path* routing algorithms [160] among tile nodes that have not been assigned to a path yet.

3) Dynamic Optimization: A fast algorithm to reduce the number of tile updates can leverage a greedy layering of current paths. For doing so, by starting from a transmitter, we keep current paths as they are up to a certain (decreasing) length and cut off the rest of the paths. Then, an algorithm can maintain performance objectives from the cutoff points, assuming them as virtual transmitters.

For minimizing the number of rounds while maintaining the loop-free update, consider the algorithm that packs as many nonconflicting tile updates into each round as possible. During each round, the algorithm starts by adding the nearest tile to the receiver that is not updated yet. From then on, it adds a new tile that is not updated to the update list if adding both of its old and updated edges to current edges does not create a cycle [156]. In most real-world scenarios, both old and updated paths between most transmitters and receivers are close to the shortest paths; therefore, the output of the greedy algorithm is not far from the optimal one. Even though there are cases in which the greedy algorithm may result in a high number of rounds, because paths between transmitters and receivers are shortest, such cases are usually rare [161]. To have a better guarantee in the worst case scenario, as we are only concerned about maintaining communication between receiver and transmitter, we can use the algorithm for loop freedom proposed in [159]. Also, in cases that a short disconnection between a receiver and a transmitter is unavoidable, we can update in two rounds: first, we update all tiles except the one that is connected to the receiver and then the tiles that are directly connected to the receiver.

VII. DISCUSSION and CHALLENGES

The complete stack of smart environment systems is presently under active investigation, and the ongoing directions and trends per layer in the literature have been outlined in Section III. The overall future challenge, however, is to create more full-stack system implementations (such as the PWEs in the case of indoor systems [5], [40]), each targeting different wireless systems: industrial [10], vehicular [162], aerial [163], medical [164], satellite [165], IoT, and sensor environments [166] are but a few of the research areas that have exhibited promising proof-of-concept gains via the integration of intelligent surfaces. Moreover, the PWE concept has been recently extended to the domain of multiphysics, bringing forth the Multiphysics-as-a-Service paradigm [167]. According to it, mechanical, acoustic, EM, and thermal propagations can be software-defined via a centralized platform in any kind of setup (e.g., even within products such as medical imaging devices) and become optimized not only for communications but also for any kind of performance objective (e.g., to increase the precision of medical imaging).

Thus, the next logical research step is the creation of complete system implementations that account for the peculiarities and needs of each environment. In this general context, we proceed to highlight the following specific challenges.

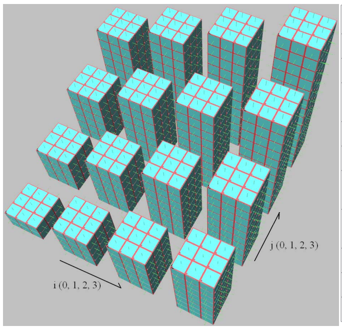
A. Cross-Layer Challenges

1) Where to Place Intelligent Surfaces?: The tile deployment within a given space is an interesting challenge that will eventually become integrated into the general network planning process. Especially in outdoor cases, where the deployment scale is large, the equipment is costly and subject to functional and legislative restrictions, and it is imperative to define fast and efficient processes for minimizing the number of employed tiles and optimize their deployment locations. Pivotal theoretical studies are already providing valuable insights in the problem [168], [169].

An exemplary process is outlined in Fig. 11. We consider an urban environment comprising of 16 buildings and one static transmitter, as shown in Fig. 11(a). The problem is to define the optimal tile locations in order to serve any single receiver placed at any random location over the road network. Multitile communications are assumed, i.e., the wireless waves can cross multiple tiles to reach the receiver.

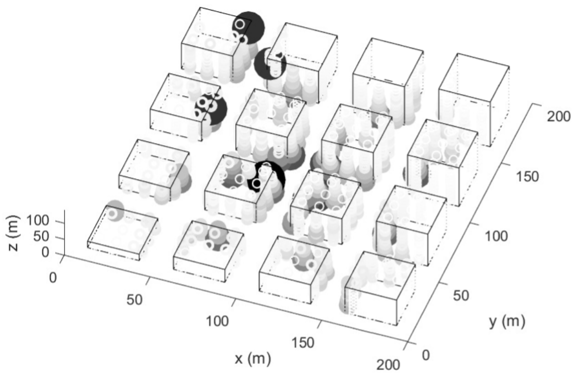
In order to draft a possible solution, we operate as follows. First, we discretize the possible tile locations over the building facades and roofs, as shown in the left inset of Fig. 11(a). Second, we assume that a tile is present and activated at each discretized location. Third, we execute a series of Monte Carlo simulations, randomizing the location of the receiver per run. In each run, we employ an intelligent environment orchestrator process, such as the KpPaths algorithm [17]. KpPaths follows the abstract graph modeling of intelligent environments described in Section V and seeks to find multiple air paths that capture the Tx emissions and guide them to the Rx. In this process, the graph model can be tuned in two ways, each defining a tile selection policy as follows.

1) The *min tiles policy* seeks to minimize the number of employed tiles per air path. As such, all edge weights in the graph are set to 1.
2) The *min intertile distance* policy employs the physical distance between tiles as the edge weights in the graph in order to minimize the overall path loss.

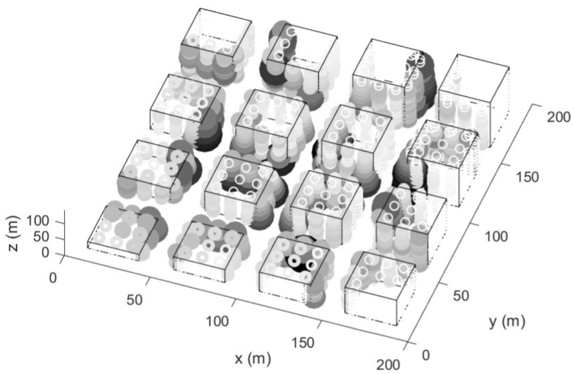


Parameter	Value
Area size	200x200m
Road width	10m
Building side (X-Y plane)	40m
Building height	2*(1+i+j)*10 m
Land	Perfect reflector
Intelligent surface functionalities	Steer, Absorb
Intelligent surface dimensions (side)	10m
Tx position	x: 100m, y: 100m
Tx power	40 dBm
Frequency	60 GHz

(a)



(b)



(c)

Fig. 11. ***Given a 3-D space, what are the optimal positions to deploy intelligent surfaces? A Monte Carlo approach is illustrated, considering multitile communications and one receiver whole location is randomized over the urban plane. Two tile selection policies are executed per run, and the usability of each tile is visualized. (a) Tile deployment optimization setup implemented in the simulation engine. (b) Tile location usability derived via the*** **min intertile distance** ***policy (darker is higher usability). (c) Tile location usability derived via the*** **min tiles** ***policy (darker is higher usability).***

We execute 100 Monte Carlo runs for each policy and obtain:

1) the set of employed tiles along with their unique identifiers, TID;
2) the received power;
3) the fraction p_{TID} of the received power crossing each of the employed tiles.

Over the course of 100 Monte Carlo runs, we aggregate the p_{TID} values, calculating the *tile usability* as $\Sigma_{100} p_{\mathrm{TID}}$, which is a simple metric to capture the usefulness of each tile location. The resulting tile usability per policy is illustrated in Fig. 11(b) and (c). Interestingly, the min tiles' configuration policy results in an even tile usability spread, while the min intertile distance policy ends up reusing a relatively small group of tiles. This is also evident from the tile usability distributions shown in Fig. 12(a). Moreover, the received power shown in Fig. 12(b) remains the same on average for both policies, but the min tiles exhibit significantly smaller variance around the median. A top-k approach applied over the tile usability distributions can yield the final, optimal tile deployment locations.

The outlined process served the purpose of showing how simple changes in a single part of the complete system can affect the tile deployment decision and give rise to interesting tradeoffs. Therefore, an interesting challenge is to design tile deployment tactics that can take into account the complete system stack and its variations. In this direction, the deployment decision should account for factors such as the physical performance limitations of the tiles from the EM aspect, the tile-to-controller internetworking latency and error rates, cross-system interference, and the presence of multiple, mobile users, and obstacles.

2) Full-Stack Security for Intelligent Environments: The role of intelligent surfaces in wireless security has been one of the first to be studied in the context of intelligent environments [9]. The avoidance of eavesdroppers (a goal well-aligned with the minimization of cross-user interference) constituted a notable goal [94], [95], [170], [171]. The concept has since been generalized as "physical layer security as an app," and it encompasses every approach for making a wireless signal physically irrecoverable around unintended recipients [23]. These approaches can be classified as application-layer security challenges, where the

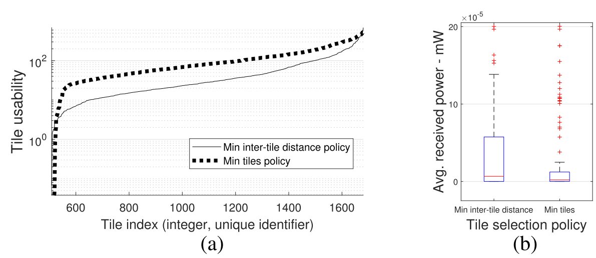


Fig. 12. ***Deployment statistics for the selection policies of Fig. 11. (a) Distribution of the tile usability per policy. (b) Received power distribution per policy.***

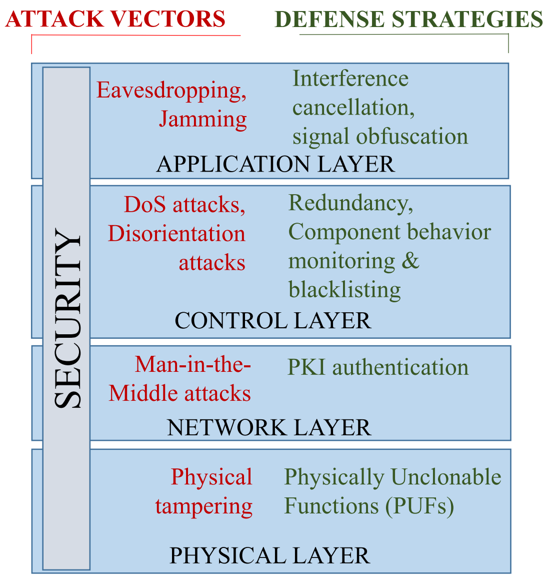


Fig. 13. ***Security challenges for intelligent environments per system layer.***

goal is to use the intelligent environment infrastructure to secure the user communications.

Nonetheless, approaches for securing the infrastructure of intelligent environments constitute a completely new and highly impactful challenge. We summarize attack vector categories and the corresponding defense strategies per system layer in Fig. 13 and denote security as a cross-layer open challenge.

At the **physical layer**, we note that intelligent surfaces will exist in the close vicinity of users, meaning that they can be susceptible to physical tampering. This can exemplary include hacks for hijacking an ISU or eavesdropping commands being exchanged with the control layer (e.g., to detect the location or status of other users). Therefore, a hardware challenge will be to include physical defenses against tampering. Physically unclonable functions (PUFs) have served this purpose in hardware in general [172]. However, the vast area of an intelligent surface, the plurality of electronic elements, and their natural susceptibility to failure constitute the integration of PUFs a novel challenge.

At the **network layer**, i.e., the networking infrastructure (switches, routers, cables, fibers, and so on) connecting the ISUs to a controller, the well-known man-in-the-middle attacks are applicable [173]. As in the physical layer, a hacker can intercept and tamper messages exchanged between the controller, and the intelligent surfaces, in order to locate devices of other users, alter their wireless performance levels, or even cause deliberate interference. Therefore, the communication between the controller and the surfaces must be secured via cryptographic means. A public key infrastructure (PKI) can be used to authenticate the controller and tile messages. Nonetheless, this raises scalability and overheard concerns, and the resulting tradeoff remains an open issue. Lightweight ciphers and IoT-based solutions constitute a promising ground for further research [174]–[176].

At the **control layer**, the central handler of all user device service requests and system events is expected to be susceptible to denial-of-service (DoS) attacks. Such congestion events can also occur from natural causes. For instance, the appearance of a flash crowd in an intelligent environment may exceed the computational capacity of the controller (or the number of deployed intelligent surfaces may not suffice). Apart from DoS, disorientation attacks are expected to be a novel attack vector for intelligent environment controllers. A hacker may program his device to beamform erratically in order to confuse the controller, forcing him to rapidly update the wireless channels, threatening their stability, and causing TCP resets. DoS defense measures include deploying redundant controllers for load balancing and defining disruption-free fail-over policies (e.g., deactivating the surfaces and reverting to natural propagation). Disorientation defense measures include the monitoring of the beamforming and mobility patterns of the users and blacklisting the ones behaving out of the norm. Related studies in SDN can provide inspiration for further research [177].

The overhead incurred by forthcoming full-stack security mechanisms in terms of added system latency will constitute a major challenge in their development. In intelligent environments, the medium under control (i.e., the wireless channel) can change abruptly and must be controlled with very tightly bounded, ultralow latency. Since the control system already adds a degree of complexity and—unavoidably—latency, a security stack must be lightweight with no efficiency compromises. In this direction, the related studies in time-sensitive and Det-Net can constitute promising starting points.

B. SDN/PWE Control System Aspects

We see a number of open research challenges with respect to the SDN control extensions of the envisioned PWE system, including the PWE system calibration, operation, and its integration with 5G/6G ecosystems. A relevant brief overview follows.

1) PWE System Calibration: The controller requires the floor plan schematics and the location of the metasurfaces in order to be able to control them, e.g., to select the most appropriate tiles to direct EM signals according to their estimated performance levels. So far, we assume that an administrator deploys metasurfaces in space, e.g., hangs them on walls, wires them to the controller, and then configures the latter with their position. This manual operation should evolve into an automated configuration process, i.e., the metasurfaces can go through a calibration phase to discover one another. This plug-and-play operation allows the metasurfaces to learn the abstracted graph model, rather than the explicit floor plan and positions. In our understanding, no tile positioning solution is currently available.

A relevant approach requires combining dynamic topology construction with in-door positioning for all involved nodes. Consequently, there is a need for relevant "geographic" routing protocols associated with recent advances in in-door positioning approaches. This should also be able to detect changes in the topology, e.g., due to a tile connectivity failure or the addition/removal of tiles. In the case of wired connectivity, redundant connections improve the reliability of the solution and should also be detected from the calibration process.

2) PWE System Operation: Here, we identified crucial requirements for the envisioned PWE system, including scalability, reliability, security, and ultralow delay constraints in the control and data channel. However, the importance of these requirements may range and depend on the considered use case. Furthermore, the type of user devices (e.g., cellular or IoT), the performance characteristics of applications running, and the requirements of the area to be covered with ISUs make the choice of technical solutions to apply challenging. In our understanding, there is a need for intelligent, logically centralized, and low-overhead control of heterogeneous technologies, which should perform in an expected manner with the given reserved resources.

Relevant research challenges on the PWE system operation include investigating the following features, all being adaptable to expressed performance goals while matching the above PWE requirements: 1) a logically centralized controller architecture implementing intent-based orchestration of end-to-end paths, involving mobile devices, ISUs, and different managed devices; 2) appropriate abstractions and interfaces that handle heterogeneous technologies, fully aligned to the abstracted graph model; 3) intelligent mechanisms that harmonize configurations of multiple technologies toward implementing end-to-end paths, including on dynamic reservation of network resources to each mobile user; 4) ultralow delay and fault-tolerant communication protocols; and 5) a flexible, low-overhead monitoring facility being bespoke to PWE end-to-end system.

3) Integration With 5G and Beyond Ecosystems: In our experience, research on PWEs focuses on their gradually improved integration with other systems and capabilities. For example, this article contributes to their better interoperation with software-defined environments. This process allows PWEs to exercise their novel performance and security capabilities, within end-to-end communication paths targeting to serve particular services or applications.

Furthermore, PWE research contributes significantly to the 5G and beyond networking vision, which targets to meet stringent performance requirements through: 1) the utilization of high-frequency bands and improved radio spectrum efficiency; 2) increased flexibility of network services through adopting novel paradigms, including SDN, network function virtualization, and edge computing; and 3) adoption of intelligent mechanisms, e.g., based on AI-/ML-based data analytics.

Along these lines, we see relevant research to evolve and cross-fertilize with PWEs, such as the following examples.

1) SDN proposals should incorporate PWEs as an important part of end-to-end flow control, i.e., bringing unique flexibility in radio communication performance and security. Relevant standardization activities are required, e.g., on PWE south interfaces and control protocols.
2) New centrally controlled time-sensitive and Det-Net approaches are important enablers of rapid radio manipulation processes that respond to changes in the network environment, e.g., a malicious user appearing in the area.
3) Edge computing can offer the processing power required for sophisticated PWE/SDN controller optimization mechanisms in close proximity to controlled ISUs. Consequently, edge resource reservation should also be part of the PWE control session establishment process.
4) New AI-/ML-based mechanisms that improve the performance and adaptability of 5G and beyond systems can also benefit PWE systems.

Last but not least, a natural evolution of our investigation is toward integrating the PWE/SDN control system into OpenRAN infrastructures, including the most prominent O-RAN architecture [178]. This allows a better alignment of PWEs with future 5G networks and beyond infrastructures. For example, O-RAN defines three types of controllers with corresponding control loops operating at different timescales: 1) the non-real-time RAN intelligent controller (non-RT RIC); 2) the near-real-time RIC (near-RT RIC); and 3) real-time schedulers operating at the distributed units (DUs), situated near the radio resources. These three controller types are usually associated with control loops having upper delay bounds of 1 s, 10 ms, and 1 ms, respectively.

Consequently, our short-term plans include implementing extensions of O-RAN architecture toward incorporating the different PWE/SDN controller components into the O-RAN controllers with respect to their delay requirements. Since we consider the end-to-end delay of the control channel to be lower than 20 ms, some components with more relaxed delay requirements (e.g., for the user association, the communication of user requirements, or QoS/QoE reporting) may reside at the near-RT RIC (or even the non-RT RIC) and those being delay-sensitive (e.g., for the manipulation of custom air paths) at real-time PWE control components (RT-PCCs) deployed near the real-time schedulers, i.e., at the DUs. The RT-PCCs should support open SDN interfaces, which may either be controlled by the O-RAN control hierarchy or implement a distributed RT-PCCs control operation (i.e., in an *ad hoc* manner). For example, in the latter case, the RT-PCCs may be rapidly and automatically switching ON/OFF specific metasurfaces, e.g., to reduce interference, as well as apply predefined "routes" among multiple placed metasurfaces.

Furthermore, the design and implementation of PWE components and interfaces should follow the directions of O-RAN toward openness and disaggregation. In this context, the system architecture should be cloud-native [179] and both external (i.e., Northbound, Southbound, and Eastbound/Westbound APIs), and internal interfaces should be defined and standardized. This allows multiple stakeholders to coexist, including network tenants, as well as meta-surface providers. The network slicing paradigm can be employed here, in a disaggregated and distributed manner, e.g., such as [180], supporting multitenancy over shared metasurface infrastructures.

As a bottom line, the design of a PWE/SDN control system involves a number of interesting research challenges, many of them being associated with similar research issues investigated in the context of 5G and beyond ecosystems. We also consider PWE systems as important technologies that contribute to the 5G and beyond networks' vision, e.g., achieving stringent performance constraints, next-generation physical layer security, and advanced in-door positioning features.

C. Theoretical Challenges

The novel structure of the models introduced by PWEs brings a set of new challenges with itself. We already discussed the algorithmic insights that can be used to address some of the challenges. However, there still exist some theoretical aspects of the problem that can be addressed in future works.

1) Modeling the Environment: Our algorithmic modeling covers most of the use cases of PWEs; however, there are still possible scenarios that can be refined in the current model. For example, PWE technologies available today cannot yet detect tile failure or tile removals. With further advancement, our models can be extended to provide efficient algorithms to detect, overcome, or even resist tile failures or removal.

2) Optimizing PWE Updates: A second major open challenge is the joint optimization of the different objectives proposed in our theoretical model. In contrast to the models usually considered in the SDN literature, in the PWE context, the new paths are often also subject to optimization. This introduces further opportunities and tradeoffs. For example, in order to compute a good tradeoff between the number of updates to be performed and the quality of the resulting paths, we could use our mixed-integer program and perform a binary search on the number of updates as a constraint.

Furthermore, some of the optimization problems that arise in PWEs reconfiguration are known to be computationally hard. For example, it is NP-hard to compute the minimum number of tile updates and also maintain performance objectives, such as power transfer maximization, as the problem reduces to finding the shortest path between shortest paths [181]. This problem is PSPACE-complete even in cases graph representing PWE is restricted [182]. As another example, it has been shown that providing a schedule for a loop-free update when any update schedule needs at least three rounds is NP-hard [159], [183]. Also, to optimize the rms delay spread, a unique loss assumption is required. Furthermore, the optimization requires relaxation to avoid nonlinear computation.

3) Data Driven Methods: From the optimization point of view, we tend to consider all possible inputs, even the worst case ones. However, the changes in a real-world environment usually follow a particular pattern. One of the ways to benefit from these patterns is to add additional assumptions about the input. To find realistic assumptions based on the data, in recent years, machine learning approaches have become popular in networking, for example, for traffic engineering [184]. There has been a first attempt to adopt neural networks for PWEs [149], but there are still many open questions on how to benefit from AI and ML to optimize our update objectives.

4) Inputs and Metrics: Providing further simulations and numerical evaluations requires realistic inputs and concrete metrics. However, previously known input generation methods that only focus on evaluating either update objectives [185] or performance objectives [149] would not fit our combined model. Furthermore, designing dedicated metrics is essential for any fair comparison of optimization results, which we consider an interesting topic for future research.

VIII. CONCLUSION

The forthcoming intelligent environments will transform the uncontrolled and typically chaotic wireless signal propagation into a deterministic, software-defined process. In this context, this study contributed to the first end-to-end system model, relating all the system components, such as the intelligent hardware, software and protocols, and user-to-environment signaling and interoperation workflows. The system model was aligned to the SDN paradigm, ensuring its direct compatibility with the existing communications infrastructure. Moreover, exploiting the logic-decoupling and abstraction properties of the SDN, this article presented a foundational algorithmic representation of the intelligent environments. A versatile graph representation of the environments transforms the problem of physical orchestration and optimization into a multi-objective path-finding problem. These abstractions of the underlying physics can facilitate the massive adoption of intelligent environments from systems engineers at large. In addition, pivotal processes for optimizing the placement of intelligent surfaces within a space were presented and supported via simulations. Finally, open challenges in security, tighter SDN integration, and analytical extensions were extensively discussed. ■

REFERENCES

[1] M. Di Renzo *et al.*, "Smart radio environments empowered by reconfigurable AI meta-surfaces: An idea whose time has come," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–20, May 2019.

[2] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "A new wireless communication paradigm through software-controlled metasurfaces," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 162–169, Sep. 2018.

[3] A. Li, S. Singh, and D. Sievenpiper, "Metasurfaces and their applications," *Nanophotonics*, vol. 7, no. 6, pp. 989–1011, Mar. 2018.

[4] X. Tan, Z. Sun, D. Koutsonikolas, and J. M. Jornet, "Enabling indoor mobile millimeter-wave networks based on smart reflect-arrays," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2018, pp. 270–278.

[5] C. Liaskos *et al.*, "The internet of metamaterial things and their software enablers," *ITU J. Future Emerg. Technol.*, vol. 1, no. 1, pp. 1–23, Nov. 2020.

[6] W. Tang *et al.*, "Wireless communications with reconfigurable intelligent surface: Path loss modeling and experimental measurement," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 421–439, Jan. 2020.

[7] H. Yang *et al.*, "A programmable metasurface with dynamic polarization, scattering and focusing control," *Sci. Rep.*, vol. 6, no. 1, pp. 1–11, Oct. 2016.

[8] A. Welkie, L. Shangguan, J. Gummeson, W. Hu, and K. Jamieson, "Programmable radio environments for smart spaces," in *Proc. 16th ACM Workshop Hot Topics Netw.*, 2017, pp. 36–42.

[9] C. Liaskos *et al.*, "A novel communication paradigm for high capacity and security via programmable indoor wireless environments in next generation wireless systems," *Ad Hoc Netw.*, vol. 87, pp. 1–16, May 2019.

[10] P. Popovski, "A primer on large intelligent surface (lis) for wireless sensing in an industrial setting," in *Proc. 15th EAI Int. Conf. (CrownCom)*, Rome, Italy, vol. 374, Nov. 2020, p. 126.

[11] Y. Pao and V. Varatharajulu, "Huygens' principle, radiation conditions, and integral formulas for the scattering of elastic waves," *J. Acoust. Soc. Amer.*, vol. 59, no. 6, pp. 1361–1371, Jun. 1976.

[12] L. Subrt and P. Pechac, "Intelligent walls as autonomous parts of smart indoor environments," *IET Commun.*, vol. 6, no. 8, p. 1004, 2012.

[13] X. Tan, Z. Sun, J. M. Jornet, and D. Pados, "Increasing indoor spectrum sharing capacity using smart reflect-array," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[14] G. Coviello, G. Avitabile, G. Piccinni, G. D'Amato, and C. Talarico, "Effects on phased arrays radiation pattern due to phase error distribution in the phase shifter operation," in *Proc. MATEC Web Conf.*, vol. 76, 2016, Art. no. 03002.

[15] A. Pitilakis *et al.*, "A multi-functional reconfigurable metasurface: Electromagnetic design accounting for fabrication aspects," *IEEE Trans. Antennas Propag.*, vol. 69, no. 3, pp. 1440–1454, Mar. 2021.

[16] D. Manessis *et al.*, "Manufacturing of high frequency substrates as software programmable metasurfaces on PCBs with integrated controller nodes," in *Proc. IEEE 8th Electron. Syst.-Integr. Technol. Conf. (ESTC)*, Sep. 2020, pp. 1–7.

[17] C. Liaskos, A. Tsioliaridou, S. Nie, A. Pitsillides, S. Ioannidis, and I. F. Akyildiz, "On the network-layer modeling and configuration of programmable wireless environments," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1696–1713, Aug. 2019.

[18] S. Abadal *et al.*, "Computing and communications for the software-defined metamaterial paradigm: A context analysis," *IEEE Access*, vol. 5, pp. 6225–6235, 2017.

[19] C. Liaskos, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "Using any surface to realize a new paradigm for wireless communications," *Commun. ACM*, vol. 61, no. 11, pp. 30–33, 2018.

[20] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "End-to-end wireless path deployment with intelligent surfaces using interpretable neural networks," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6792–6806, Nov. 2020.

[21] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. F. Akyildiz, "Mobility-aware beam steering in metasurface-based programmable wireless environments," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 9150–9154.

[22] A. B. Roach, *Session Initiation Protocol (SIP)-Specific Event Notification*, document RFC 3265, 2002.

[23] F. Mathioudakis *et al.*, "Advanced physical-layer security as an app in programmable wireless environments," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.

[24] B. Zheng and R. Zhang, "Intelligent reflecting surface-enhanced OFDM: Channel estimation and reflection optimization," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 518–522, Apr. 2020.

[25] M.-M. Zhao, Q. Wu, M.-J. Zhao, and R. Zhang, "Intelligent reflecting surface enhanced wireless networks: Two-timescale beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 2–17, Jan. 2020.

[26] Q. Wu and R. Zhang, "Joint active and passive beamforming optimization for intelligent reflecting surface assisted SWIPT under QoS constraints," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1735–1748, Aug. 2020.

[27] S. Abeywickrama, R. Zhang, Q. Wu, and C. Yuen, "Intelligent reflecting surface: Practical phase shift model and beamforming optimization," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5849–5863, Sep. 2020.

[28] X. Qian and M. D. Renzo, "Mutual coupling and unit cell aware optimization for reconfigurable intelligent surfaces," *IEEE Wireless Commun. Lett.*, vol. 10, no. 6, pp. 1183–1187, Jun. 2021.

[29] A. Roxin, J. Gaber, M. Wack, and A. Nait-Sidi-Moh, "Survey of wireless geolocation techniques," in *Proc. IEEE Globecom Workshops*, Nov. 2007, pp. 1–9.

[30] C. Liaskos *et al.*, "ABSense: Sensing electromagnetic waves on metasurfaces via ambient compilation of full absorption," in *Proc. 6th Annu. ACM Int. Conf. Nanosc. Comput. Commun.*, Sep. 2019, pp. 1–6.

[31] C. Liaskos *et al.*, "Joint compressed sensing and manipulation of wireless emissions with intelligent surfaces," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 318–325.

[32] C. De Lima *et al.*, "Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges," *IEEE Access*, vol. 9, pp. 26902–26925, 2021.

[33] A. Fascista, A. Coluccia, H. Wymeersch, and G. Seco-Granados, "RIS-aided joint localization and synchronization with a single-antenna mmwave receiver," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4455–4459.

[34] N. Engheta and R. W. Ziolkowski, *Metamaterials: Physics and Engineering Explorations*, N. Engheta and R. W. Ziolkowski, Eds. Hoboken, NJ, USA: Wiley, 2006.

[35] C. M. Soukoulis and M. Wegener, "Past achievements and future challenges in the development of three-dimensional photonic metamaterials," *Nature Photon.*, vol. 5, no. 9, pp. 523–530, Jul. 2011.

[36] S. B. Glybovski, S. A. Tretyakov, P. A. Belov, Y. S. Kivshar, and C. R. Simovski, "Metasurfaces: From microwaves to visible," *Phys. Rep.*, vol. 634, pp. 1–72, May 2016.

[37] H.-T. Chen, A. J. Taylor, and N. Yu, "A review of metasurfaces: Physics and applications," *Rep. Prog. Phys.*, vol. 79, no. 7, 2016, Art. no. 076401.

[38] A. Li, S. Singh, and D. Sievenpiper, "Metasurfaces and their applications," *Nanophotonics*, vol. 7, no. 6, pp. 989–1011, 2018.

[39] O. Tsilipakos *et al.*, "Toward intelligent metasurfaces: The progress from globally tunable metasurfaces to software-defined metasurfaces with an embedded network of controllers," *Adv. Opt. Mater.*, vol. 8, no. 17, Sep. 2020, Art. no. 2000783.

[40] The VISORSURF Project Consortium, *The Internet of Materials*, C. Liaskos, Ed. Boca Raton, FL, USA: CRC Press, 2021.

[41] K. Chen *et al.*, "A reconfigurable active Huygens' metalens," *Adv. Mater.*, vol. 29, no. 17, May 2017, Art. no. 1606422. [Online]. Available: www.researchgate.net/publication/313438142_A_Reconfigurable_Active_Huygens%27_Metalens

[42] S. Sun, Q. He, S. Xiao, Q. Xu, X. Li, and L. Zhou, "Gradient-index meta-surfaces as a bridge linking propagating waves and surface waves," *Nature Mater.*, vol. 11, no. 5, pp. 426–431, 2012.

[43] C. Pfeiffer and A. Grbic, "Controlling vector Bessel beams with metasurfaces," *Phys. Rev. Appl.*, vol. 2, no. 4, Oct. 2014, Art. no. 044012.

[44] N. Yu *et al.*, "Light propagation with phase discontinuities: Generalized laws of reflection and refraction," *Science*, vol. 334, no. 6054, pp. 333–337, Oct. 2011.

[45] F. Aieta *et al.*, "Aberration-free ultrathin flat lenses and axicons at telecom wavelengths based on plasmonic metasurfaces," *Nano Lett.*, vol. 12, no. 9, pp. 4932–4936, 2012.

[46] N. M. Estakhri and A. Alù, "Ultra-thin unidirectional carpet cloak and wavefront reconstruction with graded metasurfaces," *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 1775–1778, 2014.

[47] X. Ni, Z. J. Wong, M. Mrejen, Y. Wang, and X. Zhang, "An ultrathin invisibility skin cloak for visible light," *Science*, vol. 349, no. 6254, pp. 1310–1314, 2015.

[48] F. Liu *et al.*, "Intelligent metasurfaces with continuously tunable local surface impedance for multiple reconfigurable functions," *Phys. Rev. A, Gen. Phys.*, vol. 11, no. 4, pp. 1–10, Apr. 2019.

[49] A. Pitilakis *et al.*, "A multi-functional reconfigurable metasurface: Electromagnetic design accounting for fabrication aspects," *IEEE Trans. Antennas Propag.*, vol. 69, no. 3, pp. 1440–1454, Mar. 2021.

[50] K. M. Kossifos *et al.*, "Toward the realization of a programmable metasurface absorber enabled by custom integrated circuit technology," *IEEE Access*, vol. 8, pp. 92986–92998, 2020.

[51] C. Liaskos *et al.*, "Design and development of software defined metamaterials for nanonetworks," *IEEE Circuits Syst. Mag.*, vol. 15, no. 4, pp. 12–25, 4th Quart., 2015.

[52] D. Kouzapas *et al.*, "Towards fault adaptive routing in metasurface controller networks," *J. Syst. Archit.*, vol. 106, Jun. 2020, Art. no. 101703.

[53] (Accessed: Apr. 1, 2022). *MACOM—Product Detail—MADP-000907-14020w*. [Online]. Available: https://www.macom.com/products/product-detail/MADP-000907-14020W

[54] N. O. Parchin, H. J. Basherlou, Y. I. A. Al-Yasir, A. M. Abdulkhaleq, and R. A. Abd-Alhameed, "Reconfigurable antennas: Switching techniques—A survey," *Electronics*, vol. 9, no. 2, p. 336, Feb. 2020.

[55] S. Chenga and Z. Wu. (Accessed: Apr. 1, 2022). *Microfluidic Stretchable RF Electronics—Lab on a Chip (RSC Publishing)*. [Online]. Available: https://pubs.rsc.org/en/content/articlelanding/2010/lc/c005159d

[56] O. Quevedo-Teruel *et al.*, "Roadmap on metasurfaces," *J. Opt.*, vol. 21, no. 7, 2019, Art. no. 073002.

[57] W. E. I. Liu, Z. N. Chen, and X. Qing, "Miniature wideband non-uniform metasurface antenna using

equivalent circuit model," *IEEE Trans. Antennas Propag.*, vol. 68, no. 7, pp. 5652–5657, Jul. 2020.

[58] A. Rastgordani, Z. G. Kashani, and M. S. Abrishamian, "A general equivalent circuit model for plasmonic metasurface broadband absorbers," *Optik*, vol. 221, Nov. 2020, Art. no. 165354.

[59] M. Borgese and F. Costa, "A simple equivalent circuit approach for anisotropic frequency-selective surfaces and metasurfaces," *IEEE Trans. Antennas Propag.*, vol. 68, no. 10, pp. 7088–7098, Oct. 2020.

[60] S. V. Hum and J. Perruisseau-Carrier, "Reconfigurable reflectarrays and array lenses for dynamic antenna beam control: A review," *IEEE Trans. Antennas Propag.*, vol. 62, no. 1, pp. 183–198, Jan. 2014.

[61] D. M. Pozar, "Bandwidth of reflectarrays," *Electron. Lett.*, vol. 39, no. 21, pp. 1490–1491, Oct. 2003.

[62] L. Zhang *et al.*, "Space-time-coding digital metasurfaces," *Nat. Commun.*, vol. 9, no. 1, p. 4334, Oct. 2018.

[63] Q. Zhang, X. Wan, S. Liu, J. Y. Yin, L. Zhang, and T. J. Cui, "Shaping electromagnetic waves using software-automatically-designed metasurfaces," *Sci. Rep.*, vol. 7, no. 1, p. 3588, Dec. 2017.

[64] C.-W. Qiu, T. Zhang, G. Hu, and Y. Kivshar, "Quo vadis, metasurfaces?" *Nano Lett.*, vol. 21, no. 13, pp. 5461–5474, Jul. 2021.

[65] C. Liaskos, A. Tsioliaridou, S. Ioannidis, A. Pitsillides, and I. F. Akyildiz, "Realizing ambient backscatter communications with intelligent surfaces in 6G wireless systems," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 178–185, Feb. 2022.

[66] C. Liaskos, A. Pitilakis, A. Tsioliaridou, N. Kantartzis, and A. Pitsillides. *Initial UML Definition of the Hypersurface Compiler Middle-Ware*. European Commission Project VISORSURF: Public Deliverable D2.2. Accessed: Dec. 31, 2017. [Online]. Available: http://www.visorsurf.eu/m/VISORSURF-D2.2.pdf

[67] L. Mohjazi *et al.*, "An outlook on the interplay of artificial intelligence and software-defined metasurfaces: An overview of opportunities and limitations," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 62–73, Dec. 2020.

[68] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "End-to-end wireless path deployment with intelligent surfaces using interpretable neural networks," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6792–6806, Jul. 2020.

[69] N. Ashraf *et al.*, "Extremum seeking control for beam steering using hypersurfaces," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[70] N. Ashraf, T. Saeed, R. I. Ansari, M. Lestas, C. Liaskos, and A. Pitsillides, "Feedback based beam steering for intelligent metasurfaces," in *Proc. 2nd IEEE Middle East North Afr. Commun. Conf. (MENACOMM)*, Nov. 2019, pp. 1–6.

[71] L. Subrt and P. Pechac, "Controlling propagation environments using intelligent walls," in *Proc. 6th Eur. Conf. Antennas Propag. (EuCAP)*, 2012, pp. 1–5.

[72] M. Di Renzo *et al.*, "Smart radio environments empowered by reconfigurable AI meta-surfaces: An idea whose time has come," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–20, 2019.

[73] M. Di Renzo *et al.*, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, Jul. 2020.

[74] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, Jun. 2019.

[75] S. Hu, F. Rusek, and O. Edfors, "Beyond massive MIMO: The potential of data transmission with large intelligent surfaces," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2746–2758, May 2018.

[76] W. Qingqing and Z. Rui, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, Jan. 2019.

[77] S. Gong *et al.*, "Toward smart wireless communications via intelligent reflecting surfaces: A contemporary survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2283–2314, Jun. 2020.

[78] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Trans. Commun.*, vol. 69, no. 5, pp. 3313–3351, Jan. 2021.

[79] O. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 581–585, May 2020.

[80] B. Yigit, Y. Alapan, and M. Sitti, "Programmable collective behavior in dynamically self-assembled mobile microrobotic swarms," in *Advanced Science*, vol. 6, no. 6. Weinheim, Germany: Wiley, 2019, Art. no. 1801837.

[81] L. Wei, C. Huang, G. C. Alexandropoulos, C. Yuen, Z. Zhang, and M. Debbah, "Channel estimation for RIS-empowered multi-user MISO wireless communications," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 4144–4157, Mar. 2021.

[82] E. Basar, "Reconfigurable intelligent surface-based index modulation: A new beyond MIMO paradigm for 6G," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3187–3196, May 2020.

[83] O. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 581–585, May 2019.

[84] J. Zhang *et al.*, "Prospective multiple antenna technologies for beyond 5G," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1637–1660, Aug. 2020.

[85] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, Aug. 2019.

[86] M. Di Renzo *et al.*, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, Nov. 2020.

[87] J. Chen, Y.-C. Liang, Y. Pei, and H. Guo, "Intelligent reflecting surface: A programmable wireless environment for physical layer security," *IEEE Access*, vol. 7, pp. 82599–82612, 2019.

[88] H. Shen, W. Xu, S. Gong, Z. He, and C. Zhao, "Secrecy rate maximization for intelligent reflecting surface assisted multi-antenna communications," *IEEE Commun. Lett.*, vol. 23, no. 9, pp. 1488–1492, Sep. 2019.

[89] S. Gong *et al.*, "Toward smart wireless communications via intelligent reflecting surfaces: A contemporary survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2283–2314, Jun. 2020.

[90] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang, and D. Zhang, "A survey on green 6G network: Architecture and technologies," *IEEE Access*, vol. 7, pp. 175758–175768, 2019.

[91] W. Tang *et al.*, "Wireless communications with programmable metasurface: Transceiver design and experimental results," *China Commun.*, vol. 16, no. 5, pp. 46–61, May 2019.

[92] J. Y. Dai *et al.*, "Wireless communications through a simplified architecture based on time-domain digital coding metasurface," *Adv. Mater. Technol.*, vol. 4, no. 7, 2019, Art. no. 1900044.

[93] X. G. Zhang *et al.*, "An optically driven digital metasurface for programming electromagnetic functions," *Nature Electron.*, vol. 3, no. 3, pp. 165–171, 2020.

[94] G. Amarasuriya, R. F. Schaefer, and H. V. Poor, "Linear precoder design for physical layer security via reconfigurable intelligent surfaces," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.

[95] X. Yu, D. Xu, Y. Sun, D. W. K. Ng, and R. Schober, "Robust and secure wireless communications via intelligent reflecting surfaces," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2637–2652, Nov. 2020.

[96] F. Danufane, P. Mursia, and J. Liu, "Channel estimation in RIS-aided networks," in *Enabling 6G Mobile Networks*. Cham, Switzerland: Springer, 2022, pp. 203–220.

[97] Q.-U.-U. Nadeem, A. Kammoun, A. Chaaban, M. Debbah, and M.-S. Alouini, "Intelligent reflecting surface assisted wireless communication: Modeling and channel estimation," 2019, *arXiv:1906.02360*.

[98] J. Lin, G. Wang, R. Fan, T. A. Tsiftsis, and C. Tellambura, "Channel estimation for wireless communication systems assisted by large intelligent surfaces," 2019, *arXiv:1911.02158*.

[99] D. Mishra and H. Johansson, "Channel estimation and low-complexity beamforming design for passive intelligent surface assisted MISO wireless energy transfer," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 4659–4663.

[100] L. Wei, C. Huang, G. C. Alexandropoulos, and C. Yuen, "Parallel factor decomposition channel estimation in RIS-assisted multi-user MISO communication," in *Proc. IEEE 11th Sensor Array Multichannel Signal Process. Workshop (SAM)*, Jun. 2020, pp. 1–5.

[101] S. Khan, K. S. Khan, N. Haider, and S. Y. Shin, "Deep-learning-aided detection for reconfigurable intelligent surfaces," 2019, *arXiv:1910.09136*.

[102] A. M. Elbir, A. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "Deep channel learning for large intelligent surfaces aided mm-Wave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1447–1451, Sep. 2020.

[103] J. Chen, Y.-C. Liang, H. V. Cheng, and W. Yu, "Channel estimation for reconfigurable intelligent surface aided multi-user MIMO systems," 2019, *arXiv:1912.03619*.

[104] B. Ning, Z. Chen, W. Chen, and Y. Du, "Channel estimation and transmission for intelligent reflecting surface assisted THz communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[105] T. Saeed *et al.*, "Workload characterization of programmable metasurfaces," in *Proc. 6th Annu. ACM Int. Conf. Nanosc. Comput. Commun.*, Sep. 2019, pp. 1–6.

[106] N. M. Tran, M. M. Amri, J. H. Park, D. I. Kim, and K. W. Choi, "Reconfigurable intelligent surface-aided wireless power transfer systems: Analysis and implementation," 2021, *arXiv:2106.11805*.

[107] H. Zhao, Y. Shuang, M. Wei, T. J. Cui, P. D. Hougne, and L. Li, "Metasurface-assisted massive backscatter wireless communication with commodity Wi-Fi signals," *Nature Commun.*, vol. 11, no. 1, p. 379, Dec. 2020.

[108] Q. Jiang, G. Jin, and L. Cao, "When metasurface meets hologram: Principle and advances," *Adv. Opt. Photon.*, vol. 11, no. 3, pp. 518–576, 2019.

[109] T Saeed *et al.*, "On the use of programmable metasurfaces in vehicular networks," in *Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun.*, Sep. 2021, pp. 521–525.

[110] Q. Ma *et al.*, "Smart sensing metasurface with self-defined functions in dual polarizations," *Nanophotonics*, vol. 9, no. 10, pp. 3271–3278, Aug. 2020.

[111] L. Li *et al.*, "Intelligent metasurface imager and recognizer," *Light Sci. Appl.*, vol. 8, no. 1, p. 97, 2019.

[112] C. Liaskos, A. Tsioliaridou, S. Ioannidis, A. Pitsillides, and I. Akyildiz, "Next generation connected materials for intelligent energy propagation in multiphysics systems," *IEEE Commun. Mag.*, vol. 59, no. 8, pp. 100–106, Sep. 2021.

[113] L. Zhu *et al.*, "SDN controllers: A comprehensive

analysis and performance evaluation study," *ACM Comput. Surv.*, vol. 53, no. 6, Dec. 2020, Art. no. 133, doi: 10.1145/3421764.
[114] O. Cammaert, "Anomaly detection on power signals to monitor behavior," M.S. thesis, Dept. Inf. Technol., Fac. Eng. Archit., Ghent Univ., Ghent, Belgium, 2018.
[115] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.
[116] N. G. Nayak, F. Dürr, and K. Rothermel, "Time-sensitive software-defined network (TSSDN) for real-time applications," in *Proc. 24th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2016, pp. 193–202.
[117] L. Silva, P. Pedreiras, P. Fonseca, and L. Almeida, "On the adequacy of SDN and TSN for Industry 4.0," in *Proc. IEEE 22nd Int. Symp. Real-Time Distrib. Comput. (ISORC)*, May 2019, pp. 43–51.
[118] J.-D. Decotignie, "The many faces of industrial Ethernet [past and present]," *IEEE Ind. Electron. Mag.*, vol. 3, no. 1, pp. 8–19, Mar. 2009.
[119] N. Finn, "Introduction to time-sensitive networking," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 22–28, Jun. 2018.
[120] A. Nasrallah *et al.*, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, 1st Quart., 2019.
[121] D. Cavalcanti, J. Perez-Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, "Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems," *Proc. IEEE*, vol. 107, no. 6, pp. 1132–1152, Jun. 2019.
[122] S. Sudhakaran, K. Montgomery, M. Kashef, D. Cavalcanti, and R. Candell, "Wireless time sensitive networking for industrial collaborative robotic workcells," in *Proc. 17th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Jun. 2021, pp. 91–94.
[123] I. W. Group. (2021). *Time-Sensitive Networking (TSN) Task Group*. [Online]. Available: https://https://1.ieee802.org/tsn/
[124] D. N. (DetNet). (2021). *Deterministic Networking (DetNet) Working Group*. [Online]. Available: https://datatracker.ietf.org/wg/detnet/about/
[125] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, and H. ElBakoury, "TSN algorithms for large scale networks: A survey and conceptual comparison," 2019, *arXiv:1905.08478*.
[126] C. Ternon, J. Goossens, and J.-M. Dricot, "FTT-openFlow, on the way towards real-time SDN," *ACM SIGBED Rev.*, vol. 13, no. 4, pp. 49–54, Nov. 2016, doi: 10.1145/3015037.3015045.
[127] L. Silva, P. Goncalves, R. Marau, P. Pedreiras, and L. Almeida, "Extending OpenFlow with flexible time-triggered real-time communication services," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.
[128] K. Ahmed, J. O. Blech, M. A. Gregory, and H. Schmidt, "Software defined networking for communication and control of cyber-physical systems," in *Proc. IEEE 21st Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2015, pp. 803–808.
[129] V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107739.
[130] M. Boehm, J. Ohms, M. Kumar, O. Gebauer, and D. Wermser, "Time-sensitive software-defined networking: A unified control-plane for TSN and SDN," in *Proc. Mobile Commun. Technol. Appl. ITG-Symp.*, 2019, pp. 1–6.
[131] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven SDN controller architecture," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2014, pp. 1–6.
[132] S. Al-Rubaye, E. Kadhum, Q. Ni, and A. Anpalagan, "Industrial Internet of Things driven by SDN platform for smart grid resiliency," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 267–277, Feb. 2019.
[133] D. A. Chekired, M. A. Togou, L. Khoukhi, and A. Ksentini, "5G-slicing-enabled scalable SDN core network: Toward an ultra-low latency of autonomous driving service," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1769–1782, Aug. 2019.
[134] S. Kaur, J. Singh, and N. S. Ghumman, "Network programmability using pox controller," in *Proc. Int. Conf. Commun., Comput. Syst. (ICCCS)*, vol. 138, 2014, pp. 134–138.
[135] Y. Li, X. Su, J. Riekki, T. Kanter, and R. Rahmani, "A SDN-based architecture for horizontal Internet of Things services," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.
[136] P. Floodlight. (2021). *Floodlight*. [Online]. Available: https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller
[137] P. Berde *et al.*, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.
[138] A.-C.-G. Anadiotis, L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "Towards a software-defined network operating system for the IoT," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 579–584.
[139] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 513–521.
[140] F. Orozco-Santos, V. Sempere-Payá, T. Albero-Albero, and J. Silvestre-Blanes, "Enhancing SDN WISE with slicing over TSCH," *Sensors*, vol. 21, no. 4, p. 1075, Feb. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1075
[141] M. Baddeley *et al.*, "Atomic-SDN: Is synchronous flooding the solution to software-defined networking in IoT?" *IEEE Access*, vol. 7, pp. 96019–96034, 2019.
[142] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-defined WSN management system for IoT applications," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.
[143] T. Theodorou and L. Mamatas, "A versatile out-of-band software-defined networking solution for the Internet of Things," *IEEE Access*, vol. 8, pp. 103710–103733, 2020.
[144] T. Theodorou and L. Mamatas, "SD-MIoT: A software-defined networking solution for mobile Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4604–4617, Mar. 2021.
[145] M. Paliwal, D. Shrimankar, and O. Tembhurne, "Controllers in SDN: A review report," *IEEE Access*, vol. 6, pp. 36256–36270, 2018.
[146] B. Yan, Q. Liu, J. Shen, D. Liang, B. Zhao, and L. Ouyang, "A survey of low-latency transmission strategies in software defined networking," *Comput. Sci. Rev.*, vol. 40, May 2021, Art. no. 100386.
[147] K.-T. Foerster, S. Schmid, and S. Vissicchio, "Survey of consistent software-defined network updates," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1435–1461, Oct. 2019.
[148] W. Tang *et al.*, "On channel reciprocity in reconfigurable intelligent surface assisted wireless networks," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 94–101, Dec. 2021.
[149] C. Liaskos, A. Tsioliaridou, S. Nie, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "An interpretable neural network for configuring programmable wireless environments," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.
[150] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[151] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.
[152] P. Bosshart *et al.*, "P4: Programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014.
[153] L. Schiff, M. Borokhovich, and S. Schmid, "Reclaiming the brain: Useful OpenFlow functions in the data plane," in *Proc. 13th ACM Workshop Hot Topics Netw.*, E. Katz-Bassett, J. S. Heidemann, B. Godfrey, and A. Feldmann, Eds., Oct. 2014, pp. 7:1–7:7.
[154] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker, "Abstractions for network update," in *Proc. ACM SIGCOMM Conf.*, 2012, pp. 323–334.
[155] S. Dudycz, A. Ludwig, and S. Schmid, "Can't touch this: Consistent network updates for multiple policies," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, 2016, pp. 133–143.
[156] K. Förster, R. Mahajan, and R. Wattenhofer, "Consistent updates in software defined networks: On dependencies, loop freedom, and blackholes," in *Proc. IFIP Netw. Conf., Netw. Workshops*, Vienna, Austria, May 2016, pp. 1–9.
[157] K. Hanauer, M. Henzinger, and C. Schulz, "Recent advances in fully dynamic graph algorithms," *CoRR*, pp. 1–47, Feb. 2021.
[158] A. Madkour, W. G. Aref, F. U. Rehman, M. A. Rahman, and S. M. Basalamah, "A survey of shortest-path algorithms," *CoRR*, pp. 33–39, May 2017.
[159] K.-T. Foerster, A. Ludwig, J. Marcinkowski, and S. Schmid, "Loop-free route updates for software-defined networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 328–341, Feb. 2018.
[160] A. W. Brander and M. C. Sinclair, "A comparative study of k-shortest path algorithms," in *Performance Engineering of Computer and Telecommunications Systems*. London, U.K.: Springer, 1996, pp. 370–379.
[161] S. A. Amiri, S. Dudycz, M. Parham, S. Schmid, and S. Wiederrecht, "On polynomial-time congestion-free software-defined network updates," in *Proc. IFIP Netw. Conf.*, 2019, pp. 1–9.
[162] D. L. Dampahalage, K. S. Manosha, N. Rajatheva, and M. Latva-Aho, "Weighted-sum-rate maximization for an reconfigurable intelligent surface aided vehicular network," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 687–703, 2021.
[163] D. Ma, M. Ding, and M. Hassan, "Enhancing cellular communications for USVs via intelligent reflective surface," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.
[164] S. Eickelmann, S. Ronneberger, J. Zhang, G. Paris, and F. F. Loeffler, "Alkanes as intelligent surface thermometers: A facile approach to characterize short-lived temperature gradients on the micrometer scale," *Adv. Mater. Interfaces*, vol. 8, no. 3, Feb. 2021, Art. no. 2001626.
[165] Y.-C. Liang, R. Long, Q. Zhang, J. Chen, H. V. Cheng, and H. Guo, "Large intelligent surface/antennas (LISA): Making reflective radios smart," *J. Commun. Inf. Netw.*, vol. 42, no. 2, pp. 40–50, 2019.
[166] K. Yang, Y. Shi, Y. Zhou, Z. Yang, L. Fu, and W. Chen, "Federated machine learning for intelligent IoT via reconfigurable intelligent surface," *IEEE Netw.*, vol. 34, no. 5, pp. 16–22, Sep. 2020.
[167] C. Liaskos, A. Tsioliaridou, S. Ioannidis, A. Pitsillides, and I. F. Akyildiz, "Next generation connected materials for intelligent energy propagation in multiphysics systems," *IEEE Commun. Mag.*, vol. 59, no. 8, pp. 100–106, Aug. 2021.
[168] S. Zeng, H. Zhang, B. Di, Z. Han, and L. Song, "Reconfigurable intelligent surface (RIS) assisted wireless coverage extension: RIS orientation and location optimization," *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 269–273, Jan. 2021.
[169] G. Ghatak, "On the placement of intelligent surfaces for RSSI-based ranging in mm-wave networks," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 2043–2047, Jun. 2021.
[170] M. H. Khoshafa, T. M. N. Ngatched, and M. H. Ahmed, "Reconfigurable intelligent surfaces-aided physical layer security enhancement in D2D underlay communications," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1443–1447, May 2021.
[171] A. U. Makarfi, K. M. Rabie, O. Kaiwartya, X. Li, and R. Kharel, "Physical layer security in vehicular networks with reconfigurable intelligent surfaces,"

in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–6.
[172] F. Ganji, S. Tajik, P. Stauss, J.-P. Seifert, M. Tehranipoor, and D. Forte, "Rock'n'roll PUFs: Crafting provably secure pufs from less secure ones (extended version)," *J. Cryptograph. Eng.*, vol. 11, no. 2, pp. 105–118, 2020.
[173] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2027–2051, 3rd Quart., 2016.
[174] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers," *J. Cryptograph. Eng.*, vol. 8, no. 2, pp. 141–184, 2018.
[175] C. Manifavas, G. Hatzivasilis, K. Fysarakis, and Y. Papaefstathiou, "A survey of lightweight stream ciphers for embedded systems," *Secur. Commun. Netw.*, vol. 9, no. 10, pp. 1226–1246, Jul. 2016.
[176] E. Z. Tragos *et al.*, "Enabling reliable and secure IoT-based smart city applications," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM WORKSHOPS)*, Mar. 2014, pp. 111–116.
[177] S. W. Shin, P. Porras, V. Yegneswara, M. Fong, G. Gu, and M. Tyson, "FRESCO: Modular composable security services for software-defined networks," in *Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2013.
[178] A. Garcia-Saavedra and X. Costa-Perez, "O-RAN: Disrupting the virtualized RAN ecosystem," *IEEE Commun. Standards Mag.*, vol. 5, no. 4, pp. 96–103, Dec. 2021.
[179] 5GPPP Technology Board and 5G-IA's Trials Working Groups, White Paper. (2019). *Edge Computing for 5G Networks*. Accessed: Nov. 2021. [Online]. Available: https://bscw.5g-ppp.eu/pub/bscw.cgi/d397473/EdgeComputingFor5GNetworks.pdf
[180] S. Clayman *et al.*, "The NECOS approach to end-to-end cloud-network slicing as a service," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 91–97, Mar. 2021.
[181] M. Kamiński, P. Medvedev, and M. Milanič, "Shortest paths between shortest paths," *Theor. Comput. Sci.*, vol. 412, no. 39, pp. 5205–5210, Sep. 2011.
[182] M. Wrochna, "Reconfiguration in bounded bandwidth and treedepth," *CoRR*, vol. abs/1405.0847, pp. 1–10, May 2014.
[183] A. Ludwig, J. Marcinkowski, and S. Schmid, "Scheduling loop-free network updates: It's good to relax!" in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2015, pp. 13–22.
[184] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "Learning to route," in *Proc. 16th ACM Workshop Hot Topics Networks*, Palo Alto, CA, USA, Nov./Dec. 2017, pp. 185–191.
[185] M. Didriksen *et al.*, "Automatic synthesis of transiently correct network updates via Petri games," in *Proc. PETRI NETS*, 2021, pp. 118–137.

ABOUT THE AUTHORS

Christos Liaskos received the Diploma degree in electrical engineering from the Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece, in 2004, the M.Sc. degree in medical informatics from the Medical School, AUTH, in 2008, and the Ph.D. degree in computer networking from the Department of Informatics, AUTH, in 2014.

He is currently an Assistant Professor with the University of Ioannina, Ioannina, Greece, and an Affiliated Researcher with the Foundation for Research and Technology Hellas (FORTH), Heraklion, Greece. His research interests include computer networks, traffic engineering, and novel control schemes for wireless communications.

Lefteris Mamatas received the Diploma and Ph.D. degrees from the Department of Electrical and Computer Engineering, Democritus University of Thrace, Komotini, Greece, in 2003 and 2008, respectively.

He was a Researcher with University College London, London, U.K., the Space Internetworking Center, Democritus University of Thrace, and the DoCoMo Euro-Labs, Munich, Germany. He is currently an Associate Professor with the Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece, where he leads the Softwarized and Wireless Networks Research Group. He has published more than 70 articles in international journals and conferences. He participated in many international research projects, such as NECOS (H2020), FED4FIRE+ OC4 (H2020), WiSHFUL OC2 (H2020), MONROE OC2 (H2020), Dolfin(FP7), UniverSELF (FP7), and Extending Internet into Space (ESA). His research interests include software-defined networks, the Internet of Things, 5G networks, and multiaccess edge computing.

Dr. Mamatas has served as the General Chair of the Wired/Wireless Internet Communications (WWIC) 2016 Conference and the International Workshop on Software-Driven Flexible and Agile Networking (SWFAN) 2016 at the International Conference on Computer Communications (INFOCOM) and the TPC Chair of the International Conference on Network and Service Management (CNSM) 2022, INFOCOM SWFAN 2017, Workshop on the Emergence of Delay/Disruption-Tolerant Networks (E-DTN) 2009, and International Federation for Information Processing (IFIP) WWIC 2012 Conferences/Workshops. He has served as a Guest Editor for the *Ad Hoc Networks* (Elsevier) journal.

Arash Pourdamghani received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 2019. He is currently pursuing the Ph.D. degree with the INET Group, Technical University of Berlin, Berlin, Germany, working under the supervision of Prof. Stefan Schmid.

He was a Researcher with the University of Vienna, Vienna, Austria. He completed research internships in IST Austria, Klosterneuburg, Austria, and The Chinese University of Hong Kong (CUHK), Hong Kong.

Ageliki Tsioliaridou received the Diploma and Ph.D. degrees in electrical and computer engineering from the Democritus University of Thrace (DUTH), Komotini, Greece, in 2004 and 2010, respectively.

She is currently a Researcher with the Foundation for Research and Technology Hellas (FORTH), Heraklion, Greece. She has contributed to a number of EU, ESA, and national research projects. Her research interests lie in the area of nanonetworks, with a specific focus on architecture, protocols, and security/authorization issues.

Sotiris Ioannidis received the B.Sc. degree in mathematics and the M.Sc. degree in computer science from the University of Crete, Rethymno, Greece, in 1994 and 1996, respectively, and the Ph.D. degree from the University of Pennsylvania, Philadelphia, PA, USA, in 2005.

He was a Research Director of the Foundation for Research and Technology Hellas (FORTH), Heraklion, Greece, until 2020. He is currently an Associate Professor with the Technical University of Crete (TUC), Chania, Greece, where he is also the Director of the Microprocessor and Hardware Laboratory. He has authored more than 300 publications in international conferences, journals, and book chapters. His research interests are in the areas of systems and security.

Andreas Pitsillides (Senior Member, IEEE) is currently a Professor with the Department of Computer Science, University of Cyprus, Nicosia, Cyprus, where he is also the Co-Director of the Networks Research Laboratory (NetRL). He is also an appointed Visiting Professor with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg, South Africa, for the term 2021–2024. He has published over 340 refereed papers in flagship journals (e.g., IEEE, Elsevier, IFAC, and Springer), international conferences, and book chapters. He has coauthored two books (one edited). He participated as a principal or co-principal investigator in over 40 European Commission and locally funded research projects with over 6.7 million Euro. His broad research interests include communication networks, software-defined metasurfaces, and their application in programmable wireless environments/reconfigurable intelligent surfaces, nanonetworks, smart systems, and e-health. He has a particular interest in adapting tools from various fields of applied mathematics, such as adaptive nonlinear control theory, computational intelligence, game theory, complex systems, and nature-inspired techniques, to solve problems in communication networks.

Dr. Pitsillides received several awards, including best paper, presented keynotes, and invited lectures at major research organizations, short courses at international conferences, and short courses to industry. He has participated as the general chair and the TPC chair in many conferences.

Stefan Schmid received the M.Sc. and Ph.D. degrees from ETH Zürich, Zürich, Switzerland.

He held a postdoctoral position at the Technical University of Munich (TU Munich), Munich, Germany, and the University of Paderborn, Paderborn, Germany. In 2015, he co-founded the startup company Stacktile supported by Germany's EXIST Program, and in 2020, he helped establish the Vienna Cybersecurity and Privacy Research Center (ViSP), Faculty of Computer Science, University of Vienna, Vienna, Austria. He is currently a Professor with the Technical University of Berlin, Berlin, Germany. He is also a Senior Research Scientist with T-Labs, Berlin, an Associate Professor with Aalborg University, Aalborg, Denmark, and a Full Professor with the University of Vienna. His research interests revolve around the fundamental and algorithmic problems of networked and distributed systems.

Dr. Schmid received the IEEE Communications Society ITC Early Career Award in 2016 and the ERC Consolidator Grant in 2019.

Ian F. Akyildiz (Life Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Erlangen-Nuremberg, Nürnberg, Germany, in 1978, 1981, and 1984, respectively.

He was the Ken Byers Chair Professor of telecommunications, the Past Chair of the Telecom Group, Department of Electrical Communication Engineering (ECE), and the Director of the Broadband Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA, USA, from 1985 to 2020. He has been the Founder and the President of Truva Inc., a consulting company based in Alpharetta, GA, USA, since 1989. He has been an Adjunct Professor with the University of Cyprus, Nicosia, Cyprus, since 2017. He has been an Advisory Board Member with the Technology Innovation Institute (TII), Abu Dhabi, United Arab Emirates, since June 2020. He had many international affiliations during his career and established research centers in Spain, South Africa, Finland, Saudi Arabia, Germany, Russia, and India. In March 2022, according to Google Scholar, his H-index is 132, and the total number of citations to his articles is more than 133k. His current research interests include 6G/7G wireless systems, reconfigurable intelligent surfaces, terahertz communication, hologram communication, extended reality wireless communication, Internet of Space Things/CubeSats, Internet of Bio-Nano Things, molecular communication, underwater communication, and underground communication.

Dr. Akyildiz has been an ACM Fellow since 1997. He received numerous awards from IEEE, ACM, and other professional organizations, including the Humboldt Award from Germany and the Tubitak Award from Turkey. He has been the Founder and the Editor-in-Chief of the newly established *International Telecommunication Union Journal on Future and Evolving Technologies* (ITU J-FET) since August 2020.