IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 29, 2018, accepted September 29, 2018, date of publication October 8, 2018, date of current version October 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2873984

LayBack: SDN Management of Multi-Access Edge Computing (MEC) for Network Access Services and Radio Resource Sharing

PRATEEK SHANTHARAMA[1], AKHILESH S. THYAGATURU[2], NURULLAH KARAKOC[1], (Student Member, IEEE), LORENZO FERRARI[3], (Student Member, IEEE), MARTIN REISSLEIN[1], (Fellow, IEEE), AND ANNA SCAGLIONE[1], (Fellow, IEEE)

[1]School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287-5706, USA
[2]Software and Services Group, Intel Corporation, Chandler, AZ 85226, USA
[3]Qualcomm Technologies, Inc., San Diego, CA 92121, USA

Corresponding author: Martin Reisslein (reisslein@asu.edu)

This work was supported by the National Science Foundation through under Grant 1716121.

ABSTRACT Existing radio access networks (RANs) allow only for very limited sharing of the communication and computation resources among wireless operators and heterogeneous wireless technologies. We introduce the LayBack architecture to facilitate communication and computation resource sharing among different wireless operators and technologies. LayBack organizes the RAN communication and multi-access edge computing (MEC) resources into layers, including a devices layer, a radio node (enhanced Node B and access point) layer, and a gateway layer. LayBack positions the coordination point between the different operators and technologies just behind the gateways and thus consistently decouples the fronthaul from the backhaul. The coordination point is implemented through a software defined networking (SDN) switching layer that connects the gateways to the backhaul (core) network layer. A unifying SDN orchestrator implements an SDN-based management framework that centrally manages the fronthaul and backhaul communication and computation resources and coordinates the cooperation between different wireless operators and technologies. We illustrate the capabilities of the introduced LayBack architecture and SDN-based management framework through a case study on a novel fluid cloud RAN (CRAN) function split. The fluid CRAN function split partitions the RAN functions into function blocks that are flexibly assigned to MEC nodes, effectively implementing the RAN functions through network function virtualization. We find that for non-uniform call arrivals, the computation of the function blocks with resource sharing among operators increases a revenue rate measure by more than 25% compared to the conventional CRAN where each operator utilizes only its own resources.

INDEX TERMS 5G wireless, backhaul, fronthaul, multi-access edge computing (MEC), network function virtualization (NFV), network management, radio access network (RAN), software defined networking (SDN).

I. INTRODUCTION

Wireless access networks have emerged as a critical bottleneck in Internet access. One of the root causes of the access bottleneck is that each wireless service provider (operator) and each wireless technology (such as LTE or WiFi) operates typically in an operator/technology-specific ‘‘silo’’. That is, each operator/technology has its own radio access network (RAN) chain consisting of the RAN [1] and the corresponding backhaul network [2]. For brevity, we refer to the entire RAN chain as RANC. While there have been some efforts in wireless standards [3] and in academic research to share network resources across wireless technologies, the solutions available to date provide very limited flexibility (see Section II-A). Thus, there is only very limited statistical multiplexing (sharing) of network resources among wireless operators and technologies [4]. The *status quo* is, to a large

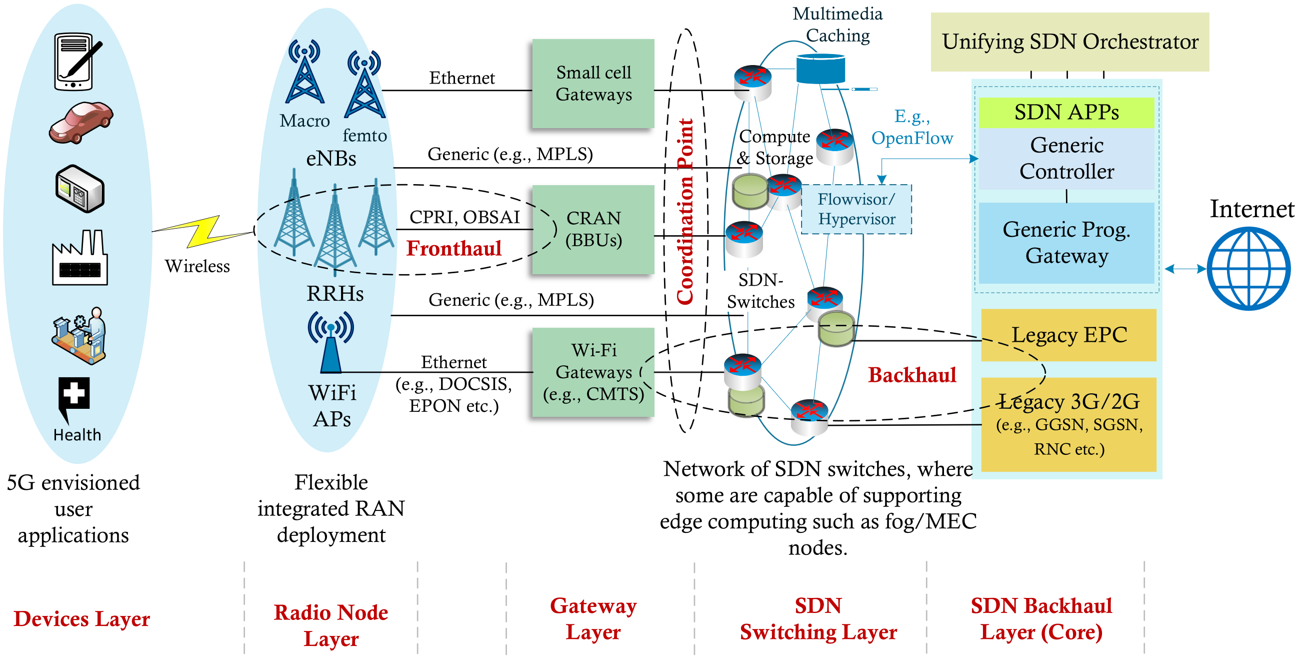


FIGURE 1. Illustration of proposed LayBack architecture: LayBack flexibly interfaces with heterogeneous radio access network (RAN) technologies through a network of gateways and SDN switches. At the "coordination point" just behind (to the right) of the respective gateways, LayBack accesses and controls the heterogeneous RANs through the SDN switching layer. The SDN switching layer consistently decouples the RAN fronthaul from the backhaul. The unifying SDN orchestrator integrates the legacy backhaul, existing architectures, and future SDN architectures. The SDN orchestrator is the central authority that controls every part of the architecture, including fronthaul and backhaul. Multi-access edge computing (MEC) nodes may be distributed throughout the radio node, gateway, SDN switching, and SDN backhaul layers.

degree, due to the lack of a convenient effective signaling infrastructure across the wireless access networks. To provide this missing signaling infrastructure, in this paper we propose a novel SDN-based architecture: the Layered Backhaul (LayBack) architecture. Our contributions are summarized next.

A. CONTRIBUTIONS

The Layered Backhaul (LayBack) architecture, which is illustrated in Fig. 1, addresses the wireless access bottleneck by judiciously employing the existing RAN and multi-access edge computing (MEC) [5] resources under a unifying Software Defined Networking (SDN) orchestrator [6], [7]. We strategically place the SDN orchestrator at the network backhaul behind the gateways of the different wireless access technologies. The centralized SDN orchestrator manages the use of MEC resources distributed across the network to provide network services, such as the RAN services.

We make three main contributions.

1) We introduce the novel LayBack architecture which comprehensively integrates the wireless fronthaul and backhaul of heterogeneous wireless technologies and operators in Section III. LayBack places the *coordination point between the heterogeneous wireless technologies and operators behind the gateways of the respective technologies and operators*. Thus, from this coordination point, an SDN switching network can flexibly interconnect the respective gateways with a unifying SDN orchestrator and the backhaul (core) networks, see Fig. 1.
2) We introduce an SDN based management framework for coordinating distributed MEC resources to support network services in Section IV. The SDN based management is executed at a unifying SDN orchestrator. The unifying SDN orchestrator performs the interlayer management and coordination within the LayBack architecture so as to readily utilize the distributed communication and computing resources across heterogeneous technologies and operators.
3) We illustrate the usage of the LayBack architecture and management framework through a quantitative case study on resource sharing in a RAN with multiple operators or technologies in Sections V and VI. The case study considers fluid RAN function splits, where RAN function block computations are dynamically assigned to MEC nodes. The evaluation results indicate that for non-uniform call arrivals, the resource sharing enabled by LayBack can increase the revenue from completed calls by more than 25%. We have presented another case study that utilizes LayBack for the optimization of communication resource allocations across different operators, gateways, and radio nodes in [8].

II. RELATED WORK

A. RAN CHAIN (RANC): FRONTHAUL AND BACKHAUL ARCHITECTURES

In contrast to clean-slate SDN-based RAN architectures, such as [9], LayBack flexibly accommodates existing as well as new technologies and deployments. The European project 5G Xhaul has studied a wide range of RANC aspects, including 5G network requirements [10] and the benefits of SDN control [11], [12]. The 5G Xhaul project also investigated aspects of specific frontend radio technologies, such as MIMO [13] and mmWave [14], and optical network technologies for the backhaul [15]. Moreover, the slicing (virtualization) of the network has been studied [16]. Similarly, the European Crosshaul project has considered the RANC combining radio fronthaul and the backhaul [17], [18]. The Crosshaul project has investigated aspects of the SDN control [19], as well as the mmWave [20] and MIMO [21] transmissions. In addition, the slicing of the network [22] and wired (including optical) transport have been considered [23]. These transport aspects are currently further examined in the European Metrohaul project [24]. Similarly, other research groups have examined slicing in RANCs [25], as well as the transport solutions for fronthaul [26] and backhaul [27]. LayBack complements the 5G Xhaul and Crosshaul architectures as well as other recently proposed SDN-based RANC architectures, such as CROWD [28], iJOIN [29], and U-WN [30], as well as similar architectures [31], [32], in that LayBack consistently decouples the wireless radio access (fronthaul) technologies, such as LTE or WiFi, and corresponding gateways from the backhaul access network.

The recently proposed SDN-based architectures generally retain some dependencies or direct interconnections between the fronthaul and the backhaul and thus have limited flexibility to accommodate heterogeneous wireless access technologies and to allow the fronthaul to evolve independently. In contrast, LayBack achieves these flexibilities by moving the management *"coordination point"* between different wireless access technologies *behind the gateways of the respective technologies*, as illustrated in Fig. 1 and elaborated in Section III. In brief, LayBack coordinates heterogeneous fronthauls and their respective gateways through central coordination behind the gateways through an SDN switching network that connects to a unifying SDN orchestrator (see Section IV). The positioning of the coordination point and the SDN switching layer just behind the respective gateways gives the LayBack SDN orchestrator direct access to the fronthauls and allows for flexible switching between heterogeneous fronthauls and backhauls.

The proposed LayBack architecture is also different from recent SDN tiered control architectures, e.g., three-tiered architectures [33], as well as prior research on SDN-based architectures, such as [12] and [15], through the tight integration of the distributed MEC with the provisioning of network access services.

B. MEC FOR RAN FUNCTION SPLITS

A computing infrastructure that is installed in close proximity to the wireless users and radio nodes is referred to as multi-access edge computing or mobile-edge computing (MEC) [34]. The MEC mechanism by Wang *et al.* [35] jointly performs user-computation offloading and radio node physical resource block (PRB) allocation (as a mechanism to manage wireless interference). Similar MEC mechanisms that jointly optimize user computations and wireless resources have been examined in [36].

Building on this prior work, we assume that computing nodes are distributed across the network. The emerging challenge is to coordinate and manage the distributed computing and network services for increasing numbers of nodes. Advanced management mechanisms are necessary, such as distributed agent-based edge computing [37] and computational resource management [38]. To the best of our knowledge, the existing fog and MEC resource management studies have been limited to the offloading of user application computations and computations for specific individual function blocks (steps) involved in wireless physical layer transmissions, e.g., interference management. Complementary to the existing mechanisms, we propose a uniform framework to comprehensively manage the computations for the full range of steps involved in providing RAN services.

The function split in RANs between the radio nodes, also referred to as remote radio heads (RRHs) or remote radio units, and the base band units (BBUs) has been investigated in several recent studies, including [39]–[42]. Our fluid RAN function split in Sections V and VI fundamentally differs from prior work in that we generalize the RAN computations to be performed via flexible function chaining [43] on distributed MEC nodes. The RAN computations are coordinated on demand through SDN control. A traditional cloud RAN (CRAN) provides the computing in a centralized manner. On the other hand, the emerging Next Generation Fronthaul Interface (NGFI) [44], [45] architecture allows for the *static* assignment of the RAN computation tasks to two specific MEC nodes, namely a Digital Unit (DU) and a Central Unit (CU), and to complete the remaining computations at the BBU. In contrast, our SDN controlled fluid RAN approach *flexibly* assigns RAN computations tasks to an arbitrary number of MEC nodes.

III. PROPOSED LAYBACK NETWORK ARCHITECTURE

LayBack is enabled by recent advances in software defined networking (SDN) [46]. LayBack breaks down the boundaries separating different wireless technologies by providing a unifying SDN-based signalling infrastructure. As shown in Fig. 1, by bringing all wireless access technologies (and corresponding operators that are willing to share their available resources and dynamic reconfiguration policies) under the umbrella of a unifying SDN orchestrator (top right of Fig. 1), LayBack achieves (*i*) the benefits of the individual wireless technologies, and (*ii*) the benefits that can be reaped through the coexistence and cooperation of multiple wireless technologies and operators.

LayBack complements and augments the potential of the popular Cloud RAN (CRAN) abstraction, because the backhaul is the point of convergence of Internet traffic and therefore is the ideal point to orchestrate the cooperative management of different wireless Internet technologies.

In traditional network infrastructures, network functions are tightly coupled with the network elements, such as the gateways, and the network elements are therefore commonly referred to as "communication nodes". In contrast in emerging MEC based infrastructures, network functions are implemented as virtualized entities on generic computing resources; hence the network elements are often referred to as "computing nodes". The LayBack architecture homogeneously considers both existing traditional and newly emerging network infrastructure deployments; thus we refer to the network elements generally as "nodes". The computing capabilities in the communication nodes in existing infrastructures can be enabled by augmenting the communication nodes with MEC nodes.

A. THE LAYERS OF THE LAYBACK ARCHITECTURE

We proceed to describe the key components and functionalities of the proposed LayBack architecture in more detail. We note that MEC nodes permeate all layers from the radio node layer to the SDN backhaul layer in Fig. 1.

1) WIRELESS END DEVICES LAYER

Mobile wireless end devices are heterogeneous and have a wide range of requirements. Providing reasonable quality-oriented services to every device is a key challenge of wireless network design. Future devices that are part of the so called Internet of Things (IoT), will likely be highly application-specific, such as health monitoring biosensors [47]. Visions for 5G wireless systems foresee that a user can request network services and applications independently of the wireless technology, i.e., physical aspects of the network connectivity, wireless protocols, and physical infrastructures of the core networks. As no single wireless technology can serve all purposes, we believe that it will be vital to provide a unifying network architecture and management framework so as to flexibly and efficiently provide wireless services.

2) RADIO NODES LAYER

Radio nodes, such as the evolved NodeB (eNB) in LTE or an access point (AP) in WiFi, provide RAN services to the end devices. Aside from LTE and WiFi, there exists a wide range of wireless access technologies (and protocols), including Wi-MAX, Zig-Bee, Bluetooth, and near field communication (NFC) [48]. These wireless technologies have unique advantages and serve unique purposes; therefore, a fluidly flexible radio node that seamlessly supports a diverse range of wireless protocols is desired [49].

RANs are not only heterogeneous in the wireless access technologies, RAN operational and deployment aspects are also highly operator specific. RAN technology advancements in the area of CRAN [39] have pushed the limits of scalability and flexibility through leveraging SDN and NFV concepts [43]. As a result of the wide range of network applications, which may be specific to operators and network architectures, the operation of the radio nodes layer is highly complex. Through our proposed LayBack architecture we can bring transparency to the network, easing the transitions among multiple heterogeneous RANs.

3) GATEWAY LAYER

The gateway layer encompasses the network entities between the radio node layer and the SDN switching layer in Fig. 1. A CRAN consists of a BBU gateway that collectively processes the basebands of several RRHs, which in turn may simultaneously support multiple wireless technologies. Radio nodes operating in a non-CRAN environment, such as non-CRAN macro cell eNBs, process the baseband locally and connect directly to the core (backhaul layer) network gateways via the SDN switching layer. Similarly, WiFi APs at residential sites typically connect to a cable or DSL modem, eventually connecting to a cable modem transmission system (CMTS) or customer premise equipment (CPE) gateway. Interactions between the gateways can be enabled by extending the gateway functions to support SDN actions, under the control of a unifying SDN orchestrator.

4) SDN SWITCHING LAYER

SDN switches are capable of a wide range of functions, such as forwarding a packet to any port, duplicating a packet on multiple ports, modifying the content inside a packet, or dropping the packet [6], [7]. The LayBack architecture homogeneously accommodates different technologies embedded in the networking switching elements. For example, a group of users who are connected to different operators, such as WiFi and LTE, can request a common content delivery service. In such a scenario, by supporting caching, the switching network elements can enable the content caching mechanism [50] serving uniformly all the users, irrespective of their wireless connectivity (i.e., LTE or WiFi) and gateway layers. The LayBack SDN switching layer directly connects to the gateways of the respective RAN technologies and operators and thus effectively provides a "coordination point" to control all RANs. At the same time, the SDN switching layer decouples the RANs (fronthaul) from the backhaul.

5) SDN BACKHAUL (CORE) NETWORK LAYER

The backhaul (core) network layer comprises technology-specific network elements, such as the Evolved Packet Core (EPC) which supports the connectivity of LTE eNBs. Similarly, for 2G/3G legacy cellular architectures, the core network includes networking elements, such as a Gateway GPRS Support Node (GGSN) and a Radio Network Controller (RNC). We define a generic programmable gateway and the SDN controller to represent all the SDN-based core network architectures, such as iJOIN and xHAUL [15]. The generic SDN controller abstracts the underlying design of the data plane and control plane specific to the architecture. The unifying SDN orchestrator extends the SDN functions to the core network elements that are not native to SDN, such as EPC and SGSN, so as to dynamically reconfigure the core network. Communication between multiple core network elements can implement the multi-operator network

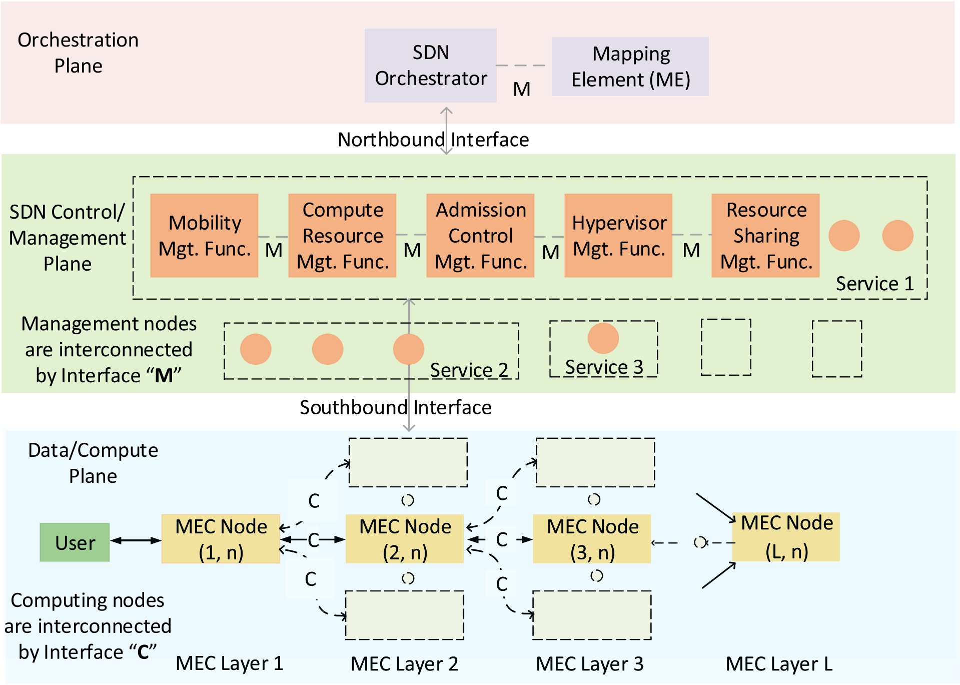


FIGURE 2. Management framework for SDN based distributed computing: The orchestration plane coordinates the overall service provisioning through instantiating control/management VMs on the control/management plane. The management plane in turn controls the data/compute plane.

sharing mechanisms as well as user mobility, e.g., handover, across multiple technologies.

6) UNIFYING SDN ORCHESTRATOR

The unifying SDN orchestrator plays an important role in creating a common platform for all the heterogeneous network technologies and operators (which can be viewed as heterogeneous network domains) across all the layers in the LayBack architecture. Although we view the SDN orchestrator as a single entity, actual orchestrator deployments can consist of multiple SDN controllers that are hierarchically organized to form a single virtual orchestrator. The unifying SDN orchestrator maintains the current topology information of the entire network and tracks the network capabilities by exchanging messages with the network elements. Network elements can either be physical entities or virtual entities obtained through NFV or network service chaining [43].

The unifying SDN orchestrator has access to all the LayBack layers to flexibly reconfigure the network. Through the central SDN orchestrator control, existing and future architectures can be flexibly integrated to achieve seamless resource sharing and mobility of users (devices) across multiple technologies. Networks maintained by different operators need to communicate their requirements and reconfiguration capabilities to the SDN orchestrator. An operator may choose not to advertise its capabilities or can selectively share capabilities based on real-time statistics, such as resource availability.

IV. SDN BASED MANAGEMENT OF DISTRIBUTED COMPUTING FOR A NETWORK SERVICE

This section introduces a management framework and the management processes to fulfill the computing requirements in a decentralized manner by dynamically reconfiguring the network based on SDN. In traditional cloud computing based networking, the computing requirements for a given user's network service are addressed in a centralized manner. Our approach not only decentralizes the computing, but also collectively delivers the distributed computing as an aggregated network service to the users.

A. MANAGEMENT FRAMEWORK PLANES AND INTERFACES

We introduce the management framework planes and interfaces illustrated in Fig. 2 for managing the provisioning of services with the LayBack architecture. In particular, we introduce from bottom to top, the data/compute plane, the control/management plane, and the orchestration plane. These planes interface with the conventional southbound and northbound interfaces of SDN. We introduce a management (M) interface for the interactions of the orchestration and control/management plane entities as well as a compute (C) interface for the interactions of the data/compute plane entities.

1) DATA/COMPUTE PLANE

The data/compute plane consists of all the SDN controlled communication and computing nodes that can be reconfigured by a logical control plane. A computing node can belong to any of the LayBack architecture layers (see Fig. 1), i.e., the RAN, gateway, switching, and core network layers.

2) CONTROL/MANAGEMENT PLANE

The control/management plane is a logical entity that is instantiated by the SDN orchestrator. More specifically, the control plane is a collection of all the management functions corresponding to the network services hosted in the data/compute plane. Essentially, the control/management plane is implemented as VMs on the MEC nodes, whereby the SDN orchestrator instantiates the management nodes, such as the SDN controller specific to a network service requested by the user. Once the control plane is provisioned, the control/management nodes (VMs) are responsible for the run-time management of the network services.

3) SDN ORCHESTRATION PLANE

The SDN orchestration plane consisting of the SDN orchestrator and the mapping element (ME, introduced in Section IV-B.3) is the logically centralized high level decision entity. In particular, the network grid is typically heterogeneous, comprising of several domains, such as, different operator and technology domains. In LayBack, the SDN orchestrator unifies these heterogeneous domains by centralizing the control decisions. The SDN orchestrator instantiates, implements, monitors, and tears down the management nodes (VMs) for the network services at the requests of users.

For inter-operator management, an operator can hide the deployment characteristics, selectively expose the deployment characteristics, or present a abstracted (virtualized) infrastructure to the centrally managed orchestrator. The SDN orchestrator then acts as the coordination point for the interaction of different network services, such as the multi-operator network sharing.

4) INTERFACES

The interactions between the various planes of the management framework and the entities within a given plane occur across pre-defined interfaces. To reduce the overhead and to ensure consistency with the general SDN management framework, conventional SDN interfaces are used for the interactions between the planes of the LayBack management framework. In particular, the interactions between the control/management plane and the data/compute plane can be supported by a conventional southbound interface, such as OpenFlow. Similarly, the interactions between the orchestration plane and the control/management plane can be supported by a conventional northbound interface, such as the representational state transfer (REST).

We introduce the management M interface for the interactions between the individual entities in the orchestration and control/management planes. Furthermore, we introduce the compute C interface for the interactions between the compute nodes in the data/compute plane. The M and C interfaces are general interface constructs that flexibly allow particular protocol interfaces to be incorporated within the general M and C interface constructs. For instance, the X2 interface (for eNB to eNB connections in LTE) or the N interface (for interconnections of network functions in the 5G backhaul) can be incorporated within the general M interface as needed to fulfill user requests. On the other hand, the S1-U interface (between eNB and S-GW in the LTE backhaul) can be incorporated into the C interface in the data/compute plane.

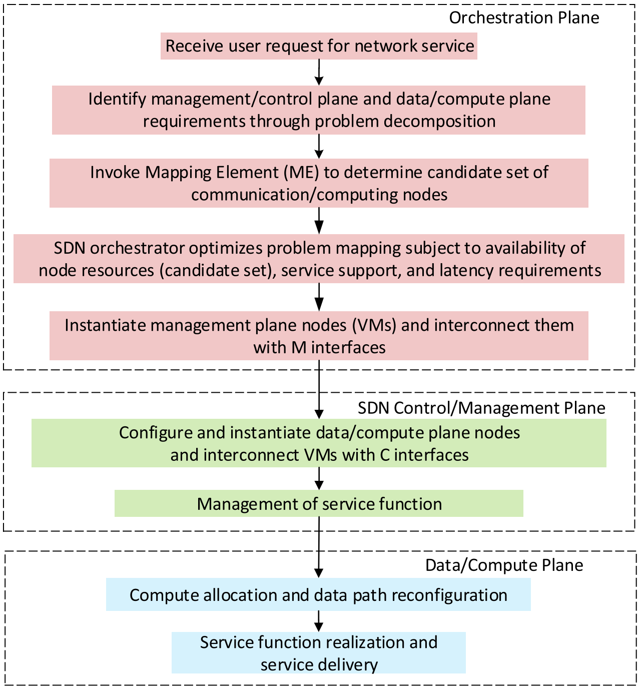

Orchestration Plane
Receive user request for network service
Identify management/control plane and data/compute plane requirements through problem decomposition
Invoke Mapping Element (ME) to determine candidate set of communication/computing nodes
SDN orchestrator optimizes problem mapping subject to availability of node resources (candidate set), service support, and latency requirements
Instantiate management plane nodes (VMs) and interconnect them with M interfaces
SDN Control/Management Plane
Configure and instantiate data/compute plane nodes and interconnect VMs with C interfaces
Management of service function
Data/Compute Plane
Compute allocation and data path reconfiguration
Service function realization and service delivery

FIGURE 3. Flow chart for SDN based management: Upon receiving a user request, the SDN orchestrator coordinates with the Mapping Element (ME) in the orchestration plane. The SDN orchestrator decomposes the problem and provisions the network connectivity among the management nodes in the SDN control/management plane to enable the control and management of the requested service. The management nodes then in turn provision the data/compute plane nodes and their interconnections in the data/compute plane for the service delivery. Overall, the SDN orchestrator is responsible for provisioning the management functions in order to achieve the end-to-end delivery of network services.

B. ORCHESTRATION LAYER PROCESSING

Adapting SDN principles [6], [7], we centralize the decision making involved in the service provisioning at the SDN orchestrator. In particular, the orchestration plane coordinates the service provisioning by executing the steps illustrated in Fig. 3 for each service request.

1) USER REQUEST FOR NETWORK SERVICE

We define a network service as a user desired network application that enables the user to interact with a remote client (cloud service) or other end-users. For instance, a network service in the 5G context could include enhanced Mobile BroadBand (eMBB), Ultra Reliable and Low Latency (URLL) communications, or a massive mobile

Internet of Things (IoT). In addition to specific applications, such as eMBB, URLL, and IoT, LayBack can also support the entire 5G framework as a network service. Thus, LayBack may provide specific network applications, such as the eMBB, URLL, and IoT, as a network service either within the framework of 5G connectivity or as independent services.

Generally, we refer to the node desiring to offload a communication or computation task arising from a service request partially or entirely to the network grid as a ‘‘user’’. We note that the ‘‘users’’ are not only the end devices, but could also include the communication and computing nodes themselves, such as, the radio nodes. A user sends the request corresponding to a network service to the network grid. The network grid forwards the request to the logically centralized SDN orchestrator. Criticality aspects of the service, such as latency and reliability requirements, are either reported or estimated based on the request type.

2) PROBLEM DECOMPOSITION FROM ORIGINAL PROBLEM TO SUB-PROBLEMS

For the purpose of management, we collectively refer to a network service or a network application as the ‘‘original problem’’, or simply as the ‘‘problem’’. Problem decomposition refers to the transformation of complex original problems into simpler constituent sub-problems preserving the problem integrity. Problem decomposition requires the consideration of the localization properties of the problem as well as the problem structure.

Many network service applications involve only a finite localized set of nodes, i.e., have specific localization properties. For instance, only the co-located radio nodes are responsible for interference coordination. Similarly, the sharing of uplink transmissions over a limited backhaul link requires the coordination of all connected users. Accordingly, for the efficient provisioning of communication and computing resources for different applications, the SDN orchestrator should consider the different sets of nodes that are co-located within a prescribed region when provisioning network services.

Depending on the problem structure, the solution of the original problem may require coordination among the sub-problems. Such coordination can be provided by a root-problem. A root-problem is a special sub-problem that is executed on a locally centralized entity that has connectivity to all the end users involved in the original problem (i.e., network application or service). In addition, the root-problem has connectivity to all other computing entities that solve sub-problems or are involved in the decision making processes. The root-problem and the individual sub-problems mutually exchange information for solving the original problem.

3) MAPPING ELEMENT: DETERMINING CANDIDATE COMMUNICATION/COMPUTING NODES SET

An important factor to consider during the problem decomposition is the availability status of the communication and computing resources. Computing entities that are part of the networking grid can simultaneously execute multiple sub-problems, in addition to their respective network functions, such as switching and forwarding. Therefore, a computing entity experiences dynamic loading based on the user requests and the current state of the network. The candidate set evaluation of the nodes needs to consider the availability of the nodes, the support for computations, the dynamic loading, the vicinity to the user, and the support for required networking services. As this involves a complex evaluation process, we propose a dedicated network Mapping Element (ME) to evaluate the candidate set of nodes for a given service request. The ME maintains and regularly updates the current states of the network nodes In particular, each node that supports communication/computing services periodically reports its utilization statistics to the ME. The ME considers the latest utilization statistics for evaluating the candidate set of nodes for a user request.

4) OPTIMIZE PROBLEM MAPPING

The SDN orchestrator employs the candidate set provided by the ME to optimize the mapping of the sub-problems (obtained from the problem decomposition) to the communication/computing nodes. More specifically, the SDN orchestrator optimizes the problem mapping subject to the node resource availability (i.e., the candidate set from the ME), the service support at the various candidate nodes, and the latency requirements.

As part of the optimization of the mapping to communication/compute nodes, the SDN orchestrator optimizes the mapping of communication services with prescribed quality of service (QoS) or quality of experience (QoE) requirements to the available access network technologies. In this communication optimization, the SDN orchestrator considers the characteristics of the different access network technologies, e.g., the different radio propagation characteristics. The specific optimization mechanisms to employ within the LayBack management framework are beyond the scope of this article and are an important direction for future research. For an initial study on optimizing communication resource allocations in the LayBack context, we refer to [8].

5) INSTANTIATE CONTROL/MANAGEMENT PLANE NODES

As a final step in its support of service provisioning, the SDN orchestrator instantiates the control/management plane nodes as VMs and interconnects the instantiated VMs with M interfaces through reconfigurable SDN switching. Alternatively, the SDN orchestrator assigns the control/management functions to existing VMs that support the required functions and have sufficient available capacity.

C. CONTROL/MANAGEMENT PLANE PROCESSING

1) INSTANTIATE DATA/COMPUTE PLANE CONFIGURATION

The control/management VMs (that were instantiated by the SDN orchestrator, see Section IV-B.5) configure the data/compute plane to instantiate and to interconnect the communication/compute nodes. More specifically, analogously to forwarding rules in an SDN switch, computing rules can be installed on the computing nodes. Each computing node is configured to process the requests if a rule pertaining to the request exists on the computing node, else the requests can be ignored, denied, or forwarded to the SDN orchestrator. Computing rules can be assigned with an expiry timeout based on the idle status of the nodes. For the typical VM based computing services, the control/managment VMs control the instantiation, migration, and tear down of data/compute plane VMs.

Moreover, the control/management VMs configure the network grid to establish the communication paths that interconnect the data/compute plane VMs. In addition, auxiliary network control functions, such as redundancy provisioning for reliability and load balancing, are conducted by the control/management VMs.

2) MAINTAIN SERVICE FUNCTIONS

Once the data/compute plane service has been instantiated, the control/managment plane maintains the service. As part of the service maintenance, the control/management plane monitors and ensures the end-to-end QoS, and preserves the service integrity in case of disruptions or network changes through recovery operations.

D. DATA/COMPUTE PLANE PROCESSING: SERVICE DELIVERY

Overall, the end-to-end service is provided through the coordinated allocation of the sub-problem communication/computation tasks to the data/compute plane nodes; whereby the data/compute plane nodes are configured by the control/management VMs. The data/compute plane nodes intercommunicate through the data paths configured via *C* interfaces by the control/management VMs. The coordinated sub-problem communication/computation actions of the data/compute plane nodes provide the overall networking services to the users.

V. LAYBACK USE CASE: NOVEL FLUID RAN FUNCTION SPLIT WITH RESOURCE SHARING ACROSS OPERATORS

The purpose of this section and the subsequent Section VI is to illustrate the use of the LayBack architecture and management framework for an exemplary use case. The exemplary use case is the provisioning of a network service through the management of distributed MEC nodes; specifically, the provisioning of a RAN service. We illustrate how the computing tasks for the RAN service can be distributed over MEC nodes and multiple operators. The distribution of the RAN service computing tasks is enabled through the management framework introduced in Section IV, which operates within the LayBack architecture introduced in Section III.

A. BACKGROUND ON EXISTING RANs

In a CRAN, an RRH is the radio frequency (RF) processing entity which is typically implemented as a part of the RF transmission antennas of cellular radio access technologies. On the other hand, the BBU performs the baseband processing. A fronthaul network interconnects the RRHs and BBUs. BBUs are softwarized entities that are typically implemented as VMs on general purpose computing entities, such as micro and macro data centers. SDN and NFV technologies can compose virtualized BBU functions through the chaining of virtualized network service functions [51]. To date, network virtualization and service chaining have been mainly applied only to the BBU functions in CRANs and to the backhaul (from BBUs toward the Internet). In contrast, we pursue network virtualization and service chaining for the RRH functions and the fronthaul (from RRH to BBU). We examine the spreading of the RRH functions across multiple layers of the LayBack architecture, while flexibly chaining function blocks together to compose efficient fronthaul links. Thus, we effectively study the extension of the benefits of VMs, NFV, and function chaining to the fronthaul.

The recently introduced generalized Next Generation Fronthaul Interface (NGFI, IEEE P1914.1) [44], [45] architecture allows for *static* functional split assignments of RAN computation tasks to the RRH and BBU as well as to two intermediate nodes, namely a Digital Unit (DU) and a Central Unit (CU). Based on the LayBack architecture and SDN based centralized management of computing for a network service, we propose a fluid RAN function split. The fluid RAN function split dynamically and flexibly assigns RAN computation tasks to arbitrary MEC nodes.

B. PROPOSED CONCEPT OF FLUID FUNCTION BLOCKS

Each function block in the NGFI fronthaul and CRAN architecture is essentially a computing entity, that transforms the incoming data to a form that is suitable for processing in the subsequent computing entity. Each computing entity may belong to a part of the radio protocol layer operations, such as PHY or MAC of LTE. In the proposed fluid function split, the CRAN function problem is partitioned into multiple sub-problems (function blocks), without a prescribed arbitrary limitation of the number of function blocks. The function blocks can be dynamically created and assigned to the computing entities, which are interconnected through Ethernet or time sensitive networking (TSN) based networks. This process not only provides a high degree of flexibility, but also facilitates new schemes for infrastructure resource utilization. NGFI limits the fronthaul function blocks to be statically split (assigned) to only two computing entities, namely the DU and the CU, in addition to the RRH and BBU. In contrast, our proposed LayBack architecture provides a unique platform for the centralized management of distributed computing so as to extend the existing fixed fronthaul and backhaul architecture to a distributed computing framework. That is, the function blocks can be flexibly assigned to distributed MEC nodes

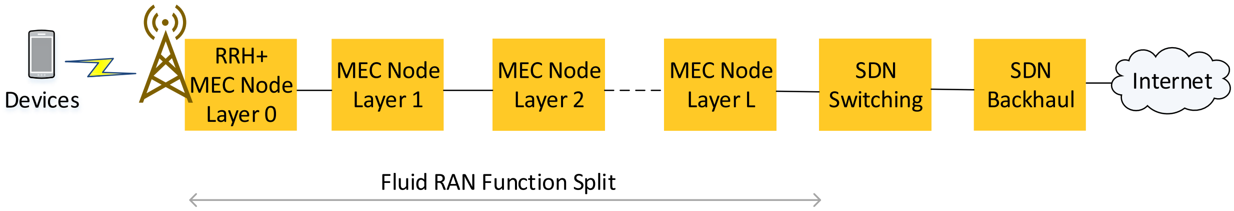


FIGURE 4. Illustration of proposed fluid RAN function split which dynamically and flexibly distributes RAN compute function blocks across multiple MEC nodes. The function blocks are chained to operate in cohesion to achieve a common function goal, i.e., to provide the RAN service. The MEC node layers $l = 0, 1, 2, \ldots, L$ are assumed to exist across the radio node layer, the fronthaul network, and the gateway layer in the overall LayBack architecture in Fig. 1.

without an arbitrary limitation on the number of utilized MEC nodes.

The fluid RAN function block assignment can be implemented through software entities, i.e., VMs, on generic computing entities. The generalized computing entities are MEC nodes distributed throughout the radio node, gateway, SDN switching layers in the LayBack architecture in Fig. 1. Existing advanced VM management methods for inter and intra data center networks [52] can be applied for the VM duplication, setup, tear down, and migration to other nodes.

C. PROPOSED LAYBACK IMPLEMENTATION OF FLUID FUNCTION SPLIT

The fundamental principle of the LayBack architecture is to unify the wide variety of heterogeneous infrastructures that exist due to different operators and technologies. LayBack categorizes these heterogeneous infrastructures in terms of layers, and interconnects them through a configurable network, i.e., the SDN switching layer, see Fig. 1. In the LayBack architecture, the RRH is located at the radio node layer, which requests services through the fluid function split paradigm. A given RRH may require different fronthaul services due to changing RRH characteristics, such as varying numbers of connected users, varying bandwidth demands, or varying power requirements. For each change in the RRH characteristics, there may be a corresponding change in the interconnecting fronthaul link requirements, and the function block implementations to complete the RAN processing.

We employ the centralized management of distributing computing, as introduced in Section IV, to meet the computing requirements for the RAN functions. More specifically, the LayBack SDN unifying orchestrator implements the SDN based management framework illustrated in Fig. 2 to assign the function blocks to MEC nodes and to configure the fronthaul network to maximize the overall utilization while seamlessly maintaining continuous service.

D. SYSTEM MODEL

1) RAN NETWORK

As summarized in Table 1, we denote N for the number of parallel CRAN systems, e.g., the number of service providers that operate a CRAN in a given area. For simplicity, we assume that each of the N parallel CRAN systems has $L + 1$ layers of MEC nodes. We denote $Z_{l,n}$ for the computation capacity at the MEC node in layer l, $0 \le l \le L$, of CRAN n, $1 \le n \le N$. We model the communication capacity of the reconfigurable SDN network interconnecting the MEC nodes as follows. The MEC nodes within a given layer l are interconnected with a shared intra-layer communication capacity C_l [bit/s]. The successive MEC layers l and $l + 1$ are interconnected by a shared inter-layer communication capacity $C_{l;l+1}$ [bit/s].

TABLE 1. Summary of main notations and parameter settings for numerical evaluations in Section VI.

	CRAN/MEC Network	
N	Number of parallel CRANs (e.g., operators)	3
L	Number of layers of MEC nodes	4
(l, n)	MEC node indices, $0 \le l \le L;\ 1 \le n \le N$	
$Z_{l,n}$	Compute capacity of node (l, n)	200
C_k	Intra-layer comm. capacity [Gbit/s] in layer k	10^3
$C_{k;l}$	Inter-layer comm. capacity [Gbit/s] betw. layers k and l	10^3
	Data Call	
r	Payload (IP level) data bitrate [Mbit/s]	5, 30, 100
τ	Expected duration [s]	2
λ	Arrival rate [calls/s] per CRAN	
	Function Blocks	
B	Number of function blocks for RAN function, indexed with $b,\ b = 0, 1, \ldots, B$	4
β_b	Computation demand (load) of function block b	Sec. VI-B
ρ_b	Bitrate [bit/s] departing function block b; $\rho_0 = R_{\mathrm{I/Q\ time}};\ \rho_B = r$	Sec. VI-B
	Performance Metrics	
O	Call blocking probability	
R	Revenue rate from completed calls	

2) DATA CALL

We define r as the payload data bitrate [in bit/s] of a given data call (stream) and let τ denote the expected (mean) call duration [in seconds]. That is, r corresponds to the user payload data rate, which we consider to be effectively the bitrate at the IP datagram level. We consider low, medium, and high user payload data rates denoted by r_{low}, r_{med}, and r_{high}. We consider independent data call generation according to a Poisson process with prescribed rate λ [data calls/s] for each of the N CRAN systems, i.e., the total call arrival rate to the N parallel CRANs is $N\lambda$.

3) FUNCTION BLOCKS

We define the function $\mathcal{F}$ to represent the complete set of of fronthaul and baseband computations for a given data call in a CRAN system. Analogous to the series expansion of any bounded function, such as the Fourier and Taylor series expansion, the CRAN function $\mathcal{F}$ can be represented in terms of function blocks as $\mathcal{F} = \sum_{b=0}^{B} f_b$, where $B + 1$ is the total number of function blocks for a given CRAN system.

In our model, a given data call (stream) has to complete the function blocks (computation tasks) $f_b,\ b = 0, 1, 2, \ldots, B$, with corresponding computation requirements (demands, loads) $\beta_b,\ b = 0, 1, 2, \ldots, B$. The function block f_0 computation has to be performed at layer $l = 0$. All other function block computations $f_b,\ b = 1, 2, \ldots, B$, can be flexibly (fluidly) performed at any of the layers $l = 1, 2, \ldots, L$.

Note that in our model, a conventional fully distributed RAN performs the function blocks $f_b,\ b = 1, 2, \ldots, B$, for a call in a given RAN in layer $l = 1$ of the RAN. That is, the computation load $\sum_{b=1}^{B} \beta_b$ is placed on layer $l = 1$ of the RAN. In contrast, in the classical CRAN scenario, the function blocks $f_b,\ b = 1, 2, \ldots, B$ are performed in layer $l = L$, i.e., at the BBU, placing computation load $\sum_{b=1}^{B} \beta_b$ on the BBU.

We denote ρ_b for the data bitrate emanating from function block f_b processing. Specifically, after function block f_0, the data bitrate is the fixed I/Q time domain data rate $\rho_0 = R_{I/Q\ time}$. Each successive function block reduces the data bitrate towards the (IP packet level) payload data rate $\rho_B = r$.

4) SERVICE POLICY

Following the optimization results for a substantial MEC load in [41], we consider an elementary greedy service policy that strives to perform the function block computations for a given call generated for CRAN m within the own CRAN m at the lowest possible layer, i.e., as close as possible to the radio nodes. The investigation of other service policies is an important direction for future research.

We consider layer $l = 0$ as a "special" layer that conducts only the essential function block f_0 that results in the time-domain I/Q stream. We do not load layer $l = 0$ with any additional computations. Instead, we greedily try to place *all remaining* function blocks $f_b,\ b = 1, 2, \ldots, B$ on node $(l = 1, m)$. If node $(l = 1, m)$ cannot accommodate this full remaining computation load $\sum_{b=1}^{B} \beta_b$, then the SDN orchestrator tries to place the maximum integral number of function blocks on the node. That is, functions $f_b,\ b = 1, 2, \ldots, \mu$, are placed on node $(l = 1, m)$ with $\mu = \{\max_{0 \le b \le B} b \text{ subject to } \sum_{a=1}^{b} \beta_a \le Z_{l=1,m}^{avail.}\}$, where $Z_{l=1,m}^{avail.}$ denotes the currently available computing capacity at node $(l = 1, m)$.

If not all $(\mu < B)$ or none $(\mu = 0)$ of the function block computation loads $\beta_b,\ b = 1, 2, \ldots, B$, can be accommodated on node $(l = 1, m)$, then the SDN orchestrator tries to move the remaining function block computations [that could not be placed on node $(l = 1, m)$] to the next higher layer, i.e., layer $l = 2$, of the same operator m, i.e., to node $(l = 2, m)$. Again, the SDN orchestrator tries to place the maximum integral number of the remaining function blocks on node $(l = 2, m)$.

If node $(l = 2, m)$ cannot accommodate all remaining function block computations, then the SDN orchestrator tries to offload the remaining function blocks to the "parallel" neighbors, i.e., to the other nodes $n \neq m,\ 1 \le n \le N$, within layer $l = 1$.

If there are still some remaining function blocks, then the SDN orchestrator tries to place these remaining computations on the next "higher" layer $l = 3$ within the own CRAN m, i.e., on node $(l = 3, m)$. Then, if there are still some remaining function blocks, the SDN orchestrator tries the other nodes $n \neq m,\ 1 \le n \le N$, in layer $l = 2$, and so on. That is, the SDN orchestrator always tries first one layer up higher in the own CRAN and if this fails, then tries the other nodes one layer back. This process continues until all nodes have been checked. Note that on the last search iteration, the SDN orchestrator cannot try to offload to layer $L + 1$ (as this layer does not exist); instead, after attempting to place the remaining function blocks on the other CRAN nodes $n \neq m,\ 1 \le n \le N$, in layer $L - 1$, the SDN orchestrator immediately proceeds to the other CRAN nodes $n \neq m,\ 1 \le n \le N$, in layer L. If some (one or more) of the functions blocks for a data call cannot be accommodated, then the call is blocked.

Throughout, the transfer of a function block from a node (k, m) to a node (l, n) requires that the data bitrate rate emanating for the call from node (k, m) can be accommodated within the currently available communication capacity out of the total intra-layer communication capacity C_k if the nodes are in the same layer $(k = l)$ or the total inter-layer communication capacity $C_{k;l}$ if the nodes are in different layers $k \neq l$. We also note that we only consider the transfer (offloading) of complete function blocks, i.e., we do not consider the splitting of a given function block f_b into sub-blocks.

5) PERFORMANCE METRICS

We evaluate the call blocking probability O for the low, medium, and high data rate calls. We evaluate the total mean revenue rate R defined as the long run average rate of completed calls weighed by the call payload data bitrate r. Moreover, we evaluate the MEC node utilization, i.e., the long-run average load level of each MEC node; in order to avoid clutter, we report the average (across the parallel N nodes in a layer) of these long-run average MEC loads for each layer $l = 1, 2, 3, 4$. We also evaluate the communication capacity utilization, i.e., the long run average bitrate transported across each of the intra-layer and inter-layer networks.

VI. FLUID RAN FUNCTION SPLIT EVALUATION

A. APPROXIMATE ANALYSIS

MEC node (l, n) can be viewed as a stochastic knapsack [53] of capacity $Z_{l,n}$. A function block f_b that is computed on node (l, n) occupies computing capacity β_b for the duration of the call. Similarly, the intra- and inter-layer communication capacities can be viewed as stochastic knapsacks. A detailed stochastic knapsack model with the different call data rates would become quite tedious. The main goal of our approximate analysis is to give insight into the sharing of the CRAN resources across the N parallel CRANs. Generally, by the scaling characteristics of stochastic knapsacks [53], one large system can support substantially more calls than a set of separate smaller systems (with the same overall capacity).

In order to derive a simple intuitive model that still captures the essential sharing dynamics, we focus on the computing aspect. We consider an approximate system model with compute capacity Z in each MEC node and one "average" call type with data bitrate $\bar{r}$ and corresponding average compute load $\bar{\beta}_b$ for function block b. In order to process an "average" call, the total computing demand $\bar{\beta}_{tot} = \sum_{b=1}^{B} \bar{\beta}_b$ has to be provided by the CRAN system. With one call type, the total CRAN compute capacity can be viewed as a classical trunking system that is characterized by the Erlang B loss formula. For a classical trunking system with a call handling capacity of Γ calls and offered load E (call arrival rate times average call holding time in Erlangs), the blocking probability is

$$O(E,\ \Gamma) = \frac{E^{\Gamma}/\Gamma!}{\sum_{\gamma=0}^{\Gamma} E^{\gamma}/\gamma!}. \tag{1}$$

In our context, a given data call requires the processing of B function blocks in the CRAN system, i.e., places a compute load $\bar{\beta}_{tot}$ on the CRAN system. The call handling capacity of one conventional CRAN system is thus $\Gamma = LZ/\bar{\beta}_{tot}$. Data calls are generated at a rate of λ call/s for a given CRAN system, whereby a given call lasts on average τ seconds. Thus, the offered load for a CRAN system is $E = \lambda\tau$. Hence, one of the stochastically identical and independent conventional CRAN systems has approximately the blocking probability $O(\lambda\tau,\ LZ/\bar{\beta}_{tot})$.

Our system with resource sharing across the N parallel CRAN systems has a total call handling capacity of $\Gamma = NLZ/\bar{\beta}_{tot}$ and a total offered load of $E = N\lambda\tau$. Thus, the blocking probability is approximately $O(N\lambda\tau,\ NLZ/\bar{\beta}_{tot})$. By the classical trunking efficiency characteristics [54], the system with resource sharing has substantially lower blocking probability, and correspondingly higher call completion rate. Accordingly, resource sharing increases the revenue rate $R = N\lambda(1 - O)\bar{r}$.

B. SIMULATION SETUP

We consider $N = 3$ parallel CRANs, each with $L + 1 = 5$ MEC node layers. We set all node computing capacities to $Z_{l,n} = 200$ [arbitrary computing units]. We set all communication capacities to $C_l = C_{k;l} = 1000$ Gbps. For each given generated call, we independently randomly select a lifetime according to an exponential distribution with mean $\tau = 2$ [s], and we uniformly randomly select a payload data bitrate r from a set of three prescribed rates, i.e., $r \in \{r_{low} = 5$ Mbps, $r_{med} = 30$ Mbps, $r_{high} = 100$ Mbps$\}$. We set the corresponding function block computing demands in the last function block $b = B = 4$ to $\beta_4^{low} = 1$, $\beta_4^{med} = 2$, and $\beta_4^{high} = 4$.

The compute loads and bitrates are typically highest for the function blocks near the radio node and decrease towards the BBU [41], [55]. We assume that each function block reduces the bitrate to a third of the bitrate entering the function block, i.e., we set $\rho_0 = R_{I/Q\ time} = 81r$, $\rho_1 = 27r$, $\rho_2 = 9r$, $\rho_3 = 3r$, and $\rho_B = \rho_4 = r$. We assume that the computing demands of the function blocks are halved for each successive function block, e.g., for a low rate call, $\beta_1 = 8$, $\beta_2 = 4$, $\beta_3 = 2$, and $\beta_4 = 1$.

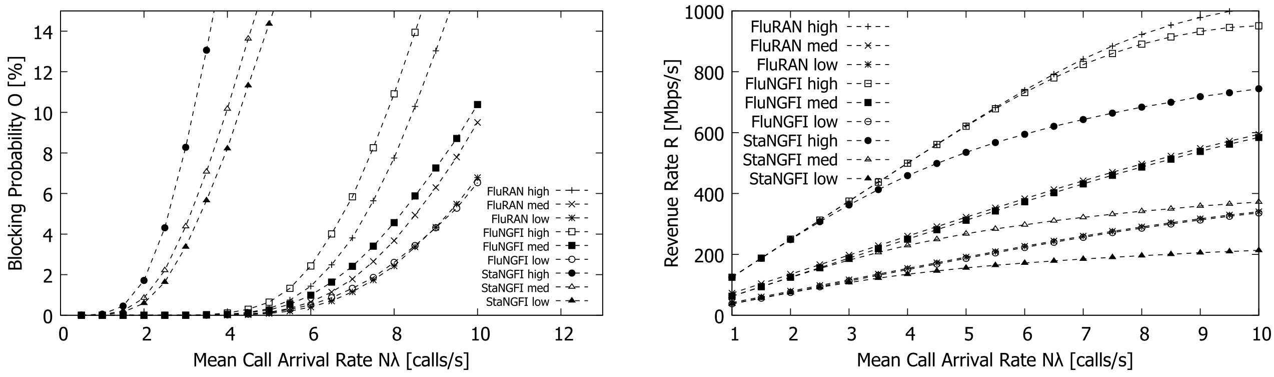
Since function block 0 is often implemented with extensive specialized hardware support, we focus on function blocks $b = 1$ through $b = B = 4$ in our evaluations. Specifically, we assume that layer $l = 0$ in each CRAN m has always enough resources to accommodate the function block 0 processing of all calls arriving to CRAN m and that bitrate $\rho_0 = 81r$ is required to offload function block $b = 1$ from node $(l = 1, m)$ to another MEC node.

We evaluate statistical confidence intervals with the batch means method. We run the simulation for a given scenario until the 95% confidence intervals for all performance metrics are less than 5% of the corresponding sample means. The confidence intervals are not plotted to avoid visual clutter.

C. EVALUATION RESULTS

1) FLUID RAN FUNCTION SPLIT

This section examines the fluid assignment of the RAN function blocks to MEC nodes. We compare our fluid RAN approach introduced in Section V with the state-of-the-art NGFI (IEEE P1914.1) based approaches, which statically assign the RAN function blocks to RRH, DU node, CU node, and BBU [39]–[42], [44], [45], [56]. Specifically, in our evaluation context, we consider the static assignment of function block f_b, $b = 1, 2, \ldots, B$, to the MEC node (b, m) of the considered CRAN m. That is, the static NGFI approach features a fine-granular splitting of the B computation tasks among the B MEC node layers; however, this fine-granular assignment is statically fixed. As an additional fluid RAN evaluation benchmark we consider a fluid NGFI which we define as follows. The function block f_0 is conducted in the DU node attached to the RRH, while the remaining function blocks f_b, $b = 1, 2, \ldots, B = 4$, with aggregate computation demand $\sum_{b=1}^{B} \beta_b$ have to be completed at one flexibly assigned MEC node. We consider the placement of this aggregate computation load according to the greedy service policy on any of the MEC nodes (l, m), $l = 1, 2, \ldots, L$, of the considered CRAN m. The assigned MEC node takes on the role of the CU for the considered call, resembling the PHY split scenario in [41]. We conduct the fluid RAN benchmark comparisons in the context of a CRAN system without resource sharing among parallel CRANs in order to bring out the performance trade-offs of the fluid (flexible) assignment of the RAN function blocks (computing tasks)



(a)

(b)

FIGURE 5. Performance of a CRAN with flexible fluid assignment of $B = 4$ RAN function block computations to $L = 4$ MEC nodes (Fluid RAN, abbreviated as FluRAN in plot, enabled by the SDN management framework in LayBack architecture), static NGFI based assignment of RAN function f_b computation to MEC node b (StaNGFI), and fluid NGFI based assignment of complete set of B RAN function computations to a MEC node out of the L MEC nodes (FluNGFI) for uniform data call arrivals to each CRAN. (a) Call Blocking Probability O. (b) Revenue Rate R.

to the MEC nodes within a given CRAN as enabled by the LayBack SDN based management framework.

We observe from Fig. 5(a) that StaNGFI has substantially higher blocking probability than FluNGFI, which in turn has slightly higher blocking probability than FluRAN. The static function block assignment with StaNGFI overloads the MEC nodes in layer $l = 1$ already for low call arrival rates with the high computation load β_1 of function block f_1. The flexible FluNGFI assignment of the complete (aggregate) set of function blocks with load $\sum_{b=1}^{B} \beta_b$ to any of the MEC nodes $l = 1, 2, 3$, or 4 avoids the overloading of the layer $l = 1$ MEC nodes. However, the complete set of function blocks requires an available capacity of at least $\sum_{b=1}^{B} \beta_b$ at a MEC node, whereas the FluRAN approach requires only at least β_1 available computing capacity at a MEC node and then enough available capacity to accommodate the other function blocks with the smaller computation loads β_2, β_3, and β_4 at the subsequent (higher indexed) MEC nodes.

We observe from Fig. 5(b) that for the practically relevant blocking probability ranges, e.g., below 5%, the revenue rates for FluRAN and FluNGFI are essentially equivalent; however, the revenue rates for StaNGFI are significantly lower. These results underscore that the flexible assignment of RAN function blocks to the MEC nodes is important for extracting high revenues from a CRAN system. On the other hand, the granularity of the function block assignment (individual function blocks with fluid RAN approach vs. aggregate of function blocks with FluNGFI) has only a relatively minor impact.

2) RAN SHARING FOR UNIFORM CALL LOAD

This section evaluates the resource sharing among CRAN systems, which LayBack enables through the positioning of the coordination point just behind the gateways of the respective RAN systems, see Fig. 1. This unique positioning of the coordination point in the LayBack architecture consistently decouples the fronthaul from the backhaul and allows for the flexible SDN control of the fronthaul and backhaul and the flexible coordination and cooperation between the different CRAN systems. In contrast, the RAN architectures in the existing literature reviewed in Section II-A generally retain some dependencies and direct interactions between fronthaul and backhaul. In these existing architectures, the fronthaul and backhaul are effectively coupled and a coordination among different RAN systems would only be possible through a coordination point behind the core networks, e.g., to the right of the legacy EPC in Fig. 1. Such a coordination point behind the core network layer would make the coordination prohibitively complex and far removed from the RAN fronthaul, i.e., would allow only for indirect control of the RAN fronthaul. Thus, fronthaul RAN sharing is generally not practical in the existing architectures. We compare a fluid RAN "no sharing" scenario representing the existing architectures (as reviewed in Section II-A) with a fluid RAN "sharing" scenario representing the LayBack architecture (whereby the sharing is among the CRANs).

In particular, we first consider a uniform call generation scenario where each of the N CRAN systems receives the same call request rate λ. The fluid RAN approach shares the resources across the N CRAN systems. In contrast, the set of N parallel "no sharing" CRAN systems do not share resources, i.e., each of the "no sharing" CRANs processes calls only within its own system. The "no sharing" CRAN system offloads function blocks according to the fluid RAN approach but only to its own MEC nodes. That is, the "no sharing" CRAN system places function blocks greedily on the own MEC nodes as close to the radio node as possible; there is no offloading to MEC nodes in parallel CRANs.

We observe from Figs. 6(a) and (b) that resource sharing among parallel CRAN systems reduces the blocking probability while increasing the revenue rate. These performance gains are due to the sharing (statistical multiplexing) of resources across a larger system with our service policy. As noted in Section VI-A, our sharing service

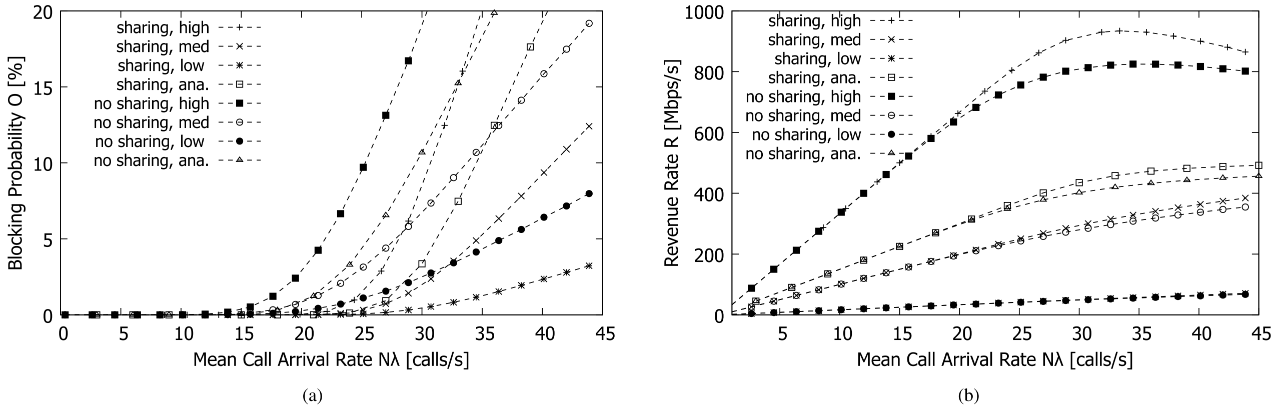


FIGURE 6. Performance of system of N parallel CRANs with resource sharing among the N CRANs (enabled by LayBack coordination point just behind RAN gateways to consistently decouple fronthaul from backhaul and to allow for SDN control of fronthaul and backhaul) vs. without resource sharing (representing conventional architectures with coupled fronthaul and backhaul that make sharing prohibitively complex, see Section II-A) for uniform data call arrivals to each CRAN. (a) Call Blocking Probability O. (b) Revenue Rate R

policy essentially lumps the N parallel CRANs into one large aggregate CRAN system with a total computing capacity of NLZ; whereas, the conventional approach has N separate CRAN systems, each with a computing capacity of LZ. The one large CRAN system obtained through the sharing can support more calls than a set of separate smaller systems (with equivalent overall capacity) due to the more flexible resource utilization in one large system compared to the separate small systems [54].

Specifically, we observe from Fig. 6(a) that for the considered uniform call load scenario, the blocking probability reduction with sharing is relatively modest, typically on the order of 5% in the critical call arrival rate range when the blocking becomes noticeable, for aggregate call arrival rates $N\lambda$ around 20 – 30 calls per second. The high rate calls require substantially more computing and communication resources than the medium and low rate calls and accordingly the high rate calls experience substantially higher blocking probabilities than the other call types. The rough analytical approximation from Section VI-A does not consider the different call types, but confirms the general trends of the blocking probability and revenue dynamics.

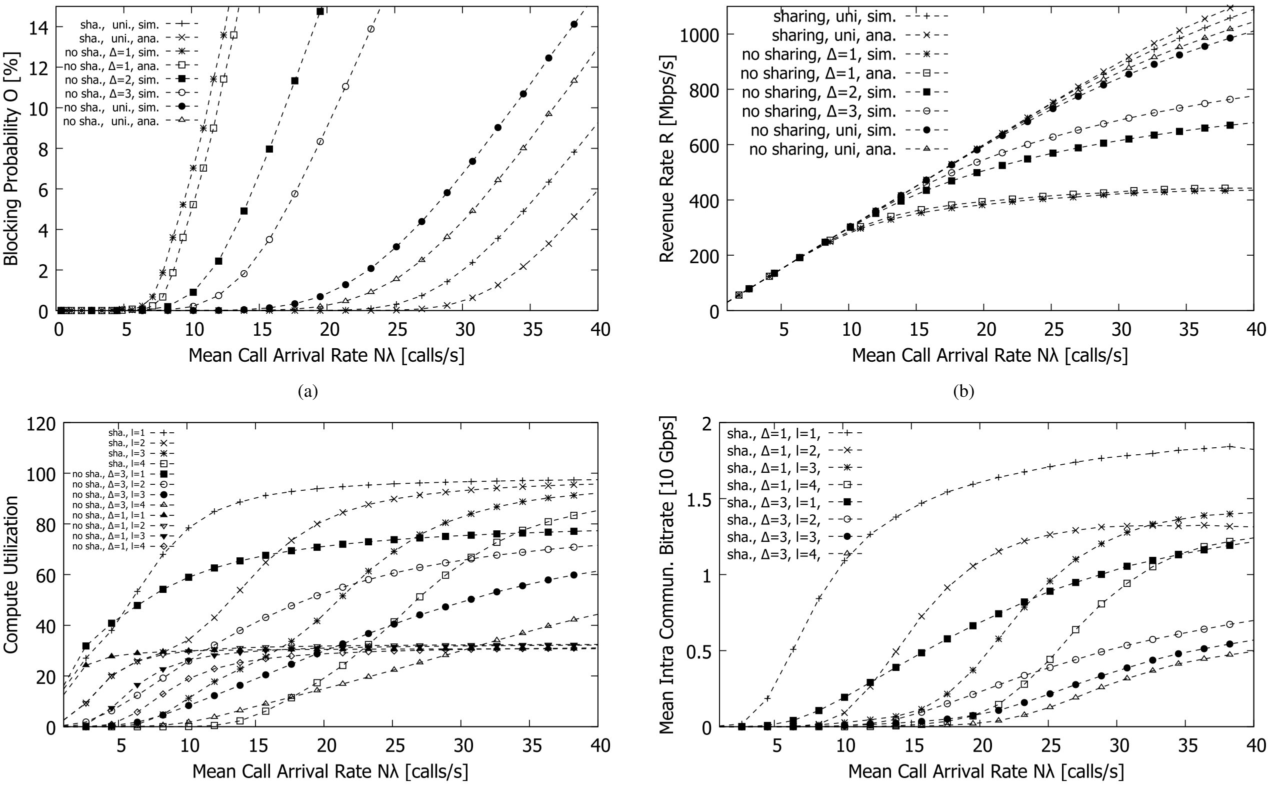
We observe from Fig. 6(b) that sharing brings only relatively small revenue increases for the uniform call load scenario. The increases with sharing are largest for the high rate calls around $N\lambda = 30 - 35$ calls/s. The high rate calls present a favorable combination of high revenue (which we set equal to the data bitrate) and low to moderate blocking probabilities up to around $N\lambda = 30 - 35$ calls/s. For higher arrival rates, more frequent low and medium rate calls fill up the free capacities and block the high rate calls, resulting in a drop of the revenue from high rate calls.

3) RAN SHARING FOR NON-UNIFORM CALL LOAD

We evaluate different skewness levels of call arrivals that may arise due to shifts in call generation, e.g., due to popular events. We consider the Zipf distribution [57] with exponent $\zeta = 1$ for different numbers Δ of CRAN systems that receive medium rate calls. In particular, for $\Delta = 1$, the entire call generation rate $N\lambda$ arrives to one CRAN system. For $\Delta = 2$, the call generation rate $N\lambda$ arrives to two CRAN systems according to the Zipf distribution with support 2, i.e., a given generated call arrives to one CRAN system with probability 2/3 and to the other CRAN system with probability 1/3. For $\Delta = 3$, the calls arrive to the three CRAN systems with proportions 6/11, 3/11, and 2/11.

We observe from Fig. 7(a) that for the CRAN system without resource sharing, the blocking probability substantially increases with increased skewness of the call arrivals, i.e., smaller Δ. We confirmed in additional simulations that are not included to avoid clutter that the blocking probability of the CRAN system with resource sharing is essentially unaffected by the skewness of the call arrivals. With sufficient intra-layer communication capacities, the resource sharing flexibly diverts the function blocks to the available resources in the parallel CRANs, keeping the call blocking low even for highly skewed call arrivals. We observe from Fig. 7(b) that the sharing greatly increases the revenue for skewed call arrivals. For the moderate case of skewed call arrivals with $\Delta = 3$ to all $N = 3$ CRAN systems, sharing can increase the revenue by approximately 25%, while more pronounced skewness in the call arrival pattern allows for even larger gains.

We also observe from Figs. 7(a) and (b) that the rough approximation with the Erlang B trunking model from Section VI-A gives slightly lower blocking probabilities than the simulated CRAN system. This is mainly because the CRAN system function blocks f_1, f_2,f_3, and f_4 have specific placement constraints that are neglected in the lumped trunking system model. Mainly, the function blocks have decreasing computing demands $\beta_1 > \beta_2 > \beta_3 > \beta_4$ and need to be executed one after the other with the placement on MEC nodes according to the considered greedy service policy. Thus, the "no sharing" CRAN system blocks for example a call if MEC layer $l = 4$ has enough free compute



(a)

(b)

(c)

(d)

FIGURE 7. Performance of system of *N* parallel CRANs with resource sharing among the *N* CRANs (enabled by LayBack) vs. without resource sharing (conventional architectures with prohibitive sharing complexity) for non-uniform arrivals of medium rate data calls according to Zipf distribution to Δ CRANs. (a) Call Blocking Probability *O*. (b) Revenue Rate *R*. (c) MEC Utilization. (d) Intra-layer Communication Bitrates with Sharing.

capacity for β_1, but not enough to accommodate $\sum_{b=1}^{B} \beta_b$, and the other MEC layers $l = 1, 2$, and 3 have enough free capacity for β_2, β_3, and β_4, but not enough for β_1; while this example call would be accommodated in the trunking system.

For the compute utilization, we observe from Fig. 7(c) that without sharing the utilization levels are quite low compared to the CRANs with sharing. Without sharing, the $\Delta = 1$ scenario already fully loads the resources in the one CRAN receiving all the calls for fairly low call arrival rates. The resources in the two parallel CRANs cannot be utilized, i.e., they have a utilization of zero. Thus, the overall utilization of the compute resources of the $N = 3$ parallel CRANs is limited to one third. In contrast, the sharing utilizes the compute resources across all $N = 3$ parallel CRANs. (Additional simulations that are not included to avoid clutter confirmed that sharing achieves very similar utilization levels for all considered call arrival patterns.) The considered greedy function block placement policy more and more fully utilizes the successive MEC layers $l = 1, 2, 3$, and 4 as the call arrivals increase. For the $\Delta = 3$ scenario in Fig. 7(c), all $N = 3$ CRANs receive calls, but with rates skewed according to the Zipf distribution. Without sharing, the CRAN system receiving the highest proportion of calls reaches near full utilization already for moderate call arrival rates and blocks calls, while the two parallel CRAN systems have still unutilized compute resources. In contrast, the fluid RAN system with resource sharing among the N CRAN systems consistently achieves high resource utilization, low blocking probability, and high revenue for a wide range of call arrival patterns.

Fig. 7(d) shows the intra-layer communication bitrates for layers $l = 1, 2, 3$, and 4 for the resource sharing between the $N = 3$ parallel CRAN systems. We observe that the extreme case of all calls arriving to $\Delta = 1$ CRAN system results in relatively high bitrates in layer $l = 1$ already for low call arrival rates. The intra-layer bitrates in the successive layers $l = 2, 3$, and 4 increase as the call arrival rate increases. This behavior is in accordance with the considered greedy service policy that strives to complete function blocks in the lowest indexed layers. For the more realistic case of skewed arrivals to all $\Delta = N = 3$ CRAN systems, we observe significantly lower intra-layer communication bitrates, whereby layer $l = 1$ experiences again the highest intra-layer bitrates.

VII. CONCLUSIONS

We have introduced the Layered Backhaul (LayBack) architecture for coordinating heterogeneous radio access networks (RANs) with software defined networking (SDN).

LayBack ties the heterogeneous RANs together behind their respective gateways, such as cloud RAN (CRAN) baseband units of small cell gateways. More specifically, these heterogeneous gateways are connected by an SDN network to a unifying SDN orchestrator. We have introduced an SDN based management framework that is executed in the SDN orchestrator. The management framework coordinates distributed computing resources, such as distributed multiple-access edge computing (MEC) nodes, to cohesively provide computing for network services.

We showcased the LayBack architecture and management framework for a novel fluid cloud RAN (CRAN) function split. The fluid function split partitions the entire set of CRAN fronthaul computations into multiple function blocks. The function blocks are assigned to MEC nodes according to a service policy. We evaluated an elementary greedy service policy that shares resources between the CRAN systems of different operators. We found that the resource sharing substantially increases the revenue rate from the CRAN service.

LayBack can serve as basis for a wide range of future research directions. One direction is to develop and evaluate optimization mechanisms for the function splitting (problem decomposition) and the allocation of the resulting function blocks (sub-problems). Initial work in this direction has, for instance, explored time-scale based decompositions [8], [58]. The convergence and optimality characteristics of such time-scale decompositions as well as other problem decomposition approaches need to be thoroughly examined in future research.

REFERENCES

[1] E. Pateromichelakis *et al.*, "Service-tailored user-plane design framework and architecture considerations in 5G radio access networks," *IEEE Access*, vol. 5, pp. 17089–17105, 2017.

[2] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "5G backhaul challenges and emerging research directions: A survey," *IEEE Access*, vol. 4, pp. 1743–1766, 2016.

[3] T.-C. Liu, K. Wang, C.-Y. Ku, and Y.-H. Hsu, "QoS-aware resource management for multimedia traffic report systems over LTE-A," *Comput. Netw.*, vol. 94, pp. 375–389, Jan. 2016.

[4] B. Niu, Y. Zhou, H. Shah-Mansouri, and V. W. Wong, "A dynamic resource sharing mechanism for cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 12, pp. 8325–8338, Dec. 2016.

[5] W. Fan, Y. Liu, B. Tang, F. Wu, and Z. Wang, "Computation offloading based on cooperations of mobile edge computing-enabled base stations," *IEEE Access*, vol. 6, pp. 22622–22633, 2018.

[6] W. Huang, L. Ding, D. Meng, J.-N. Hwang, Y. Xu, and W. Zhang, "QoE-based resource allocation for heterogeneous multi-radio communication in software-defined vehicle networks," *IEEE Access*, vol. 6, pp. 3387–3399, 2018.

[7] O. Narmanlioglu, E. Zeydan, and S. S. Arslan, "Service-aware multi-resource allocation in software-defined next generation cellular networks," *IEEE Access*, vol. 6, pp. 20348–20363, 2018.

[8] L. Ferrari, N. Karakoc, A. Scaglione, M. Reisslein, and A. Thyagaturu, "Layered cooperative resource sharing at a wireless SDN backhaul," in *Proc. IEEE ICC Workshops*, May 2018, pp. 1–6.

[9] P. Ameigeiras, J. J. Ramos-Munoz, L. Schumacher, J. Prados-Garzon, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Link-level access cloud architecture design based on SDN for 5G networks," *IEEE Netw.*, vol. 29, no. 2, pp. 24–31, Mar./Apr. 2015.

[10] J. Bartelt *et al.*, "5G transport network requirements for the next generation fronthaul interface," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, p. 89, Dec. 2017.

[11] J. Gutiérrez *et al.*, "5G-XHaul: A converged optical and wireless solution for 5G transport networks," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 9, pp. 1187–1195, Sep. 2016.

[12] A. De La Oliva *et al.*, "Xhaul: Toward an integrated fronthaul/backhaul architecture in 5G networks," *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 32–40, Oct. 2015.

[13] J. K. Chaudhary, J. Bartelt, and G. Fettweis, "Statistical multiplexing in fronthaul-constrained massive MIMO," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–6.

[14] D. Huerfano, I. Demirkol, and P. Legg, "Joint optimization of path selection and link scheduling for millimeter wave transport networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 115–120.

[15] A. Tzanakaki *et al.*, "Wireless-optical network convergence: Enabling the 5G architecture to support operational and end-user services," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 184–192, Oct. 2017.

[16] S. Costanzo, I. Fajjari, N. Aitsaadi, and R. Langar, "A network slicing prototype for a flexible cloud radio access network," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–4.

[17] F. Cavaliere *et al.*, "Towards a unified fronthaul-backhaul data plane for 5G the 5G-Crosshaul project approach," *Comp. Standards Interfaces*, vol. 51, pp. 56–62, Mar. 2017.

[18] X. Costa-Perez *et al.*, "5G-Crosshaul: An SDN/NFV integrated fronthaul/backhaul transport network architecture," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 38–45, Feb. 2017.

[19] S. González *et al.*, "5G-Crosshaul: An SDN/NFV control and data plane architecture for the 5G integrated Fronthaul/Backhaul," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 9, pp. 1196–1205, Sep. 2016.

[20] H. Ogawa, G. K. Tran, K. Sakaguchi, and T. Haustein, "Traffic adaptive formation of mmWave meshed backhaul networks," in *Proc. IEEE ICC Workshops*, May 2017, pp. 185–191.

[21] Y. Huang, C. Lu, M. Berg, and P. Ödling, "Functional split of zero-forcing based massive MIMO for fronthaul load reduction," *IEEE Access*, vol. 6, pp. 6350–6359, 2018.

[22] X. Li *et al.*, "5G-crosshaul network slicing: Enabling multi-tenancy in mobile transport networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 128–137, Aug. 2017.

[23] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, "Toward an efficient C-RAN optical fronthaul for the future networks: A tutorial on technologies, requirements, challenges, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 708–769, 1st Quart., 2018.

[24] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, "Control, management, and orchestration of optical networks: Evolution, trends, and challenges," *J. Lightw. Technol.*, vol. 36, no. 7, pp. 1390–1402, Apr. 1, 2018.

[25] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Trans. Netw. Serv. Manage.*, vol. 13, no. 3, pp. 462–476, Sep. 2016.

[26] P. Chanclou, L. A. Neto, K. Grzybowski, Z. Tayq, F. Saliou, and N. Genay, "Mobile fronthaul architecture and technologies: A RAN equipment assessment [invited]," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 1, pp. A1–A7, Jan. 2018.

[27] A. S. Thyagaturu, Y. Dashti, and M. Reisslein, "SDN-based smart gateways (Sm-GWs) for multi-operator small cell network management," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 740–753, Dec. 2016.

[28] S. Auroux, M. Dräxler, A. Morelli, and V. Mancuso, "Dynamic network reconfiguration in wireless DenseNets with the CROWD SDN architecture," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2015, pp. 144–148.

[29] D. Wang, L. Zhang, Y. Qi, and A. Ul Quddus, "Localized mobility management for SDN-integrated LTE backhaul networks," in *Proc. IEEE VTC*, May 2015, pp. 1–6.

[30] S. Zhang, C. Kai, and L. Song, "SDN based uniform network architecture for future wireless networks," in *Proc. Int. Conf. Comput., Commun. Netw. Techn. (ICCCNT)*, Jul. 2014, pp. 1–5.

[31] H. Droste *et al.*, "An adaptive 5G multiservice and multitenant radio access network architecture," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 9, pp. 1262–1270, 2016.

[32] J. Qadir, N. Ahmed, and N. Ahad, "Building programmable wireless networks: An *architectural survey*," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, p. 172, 2014.

[33] I. Elgendi, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "Traffic offloading techniques for 5G cellular: A three-tiered SDN architecture," *Ann. Telecommun.*, vol. 71, nos. 11–12, pp. 583–593, Dec. 2016.

[34] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[35] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[36] P. Luong, F. Gagnon, C. Despins, and L.-N. Tran, "Joint virtual computing and radio resource allocation in limited fronthaul green C-RANs," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2602–2617, Apr. 2018.

[37] V. K. C. Bumgardner, V. W. Marek, and C. D. Hickey, "Cresco: A distributed agent-based edge computing framework," in *Proc. IEEE Int. Conf. Netw. Service Manage.*, Oct./Nov. 2016, pp. 400–405.

[38] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[39] A. Checko *et al.*, "Cloud RAN for mobile networks—A technology overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2015.

[40] C.-L. I, "Seven fundamental rethinking for next-generation wireless communications," *APSIPA Trans. Signal Inf. Process.*, vol. 6, p. e10, Sep. 2017.

[41] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis, "FluidRAN: Optimized vRAN/MEC orchestration," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1–9.

[42] A. S. Thyagaturu, Z. Alharbi, and M. Reisslein, "R-FFT: Function split at IFFT/FFT in unified LTE CRAN and cable access network," *IEEE Trans. Broadcast.*, vol. 64, no. 3, pp. 648–665, Sep. 2018.

[43] A. Leivadeas, G. Kesidis, M. Falkner, and I. Lambadaris, "A graph partitioning game theoretical approach for the VNF service chaining problem," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 890–903, Dec. 2017.

[44] IEEE. *Next Generation Fronthaul Interface (1914) Working Group*. Acessed: Oct. 5, 2018. [Online]. Available: http://sites.ieee.org/sagroups-1914

[45] C.-L. I, H. Li, J. Korhonen, J. Huang, and L. Han, "RAN revolution with NGFI (xHaul) for 5G," *J. Lightw. Technol.*, vol. 36, no. 2, pp. 541–550, Jan. 15, 2018.

[46] C. Ramirez-Perez and V. Ramos, "SDN meets SDR in self-organizing networks: Fitting the pieces of network management," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 48–57, Jan. 2016.

[47] J. J. P. C. Rodrigues *et al.*, "Enabling technologies for the Internet of Health Things," *IEEE Access*, vol. 6, pp. 13129–13141, 2018.

[48] H.-J. Kim, H. Hirayama, S. Kim, K. J. Han, R. Zhang, and J.-W. Choi, "Review of near-field wireless power and communication for biomedical applications," *IEEE Access*, vol. 5, pp. 21264–21285, 2017.

[49] K. Sundaresan, M. Y. Arslan, S. Singh, S. Rangarajan, and S. V. Krishnamurthy, "FluidNet: A flexible cloud-based radio access network for small cells," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 915–928, Apr. 2016.

[50] N. Zhao, X. Liu, F. R. Yu, M. Li, and V. C. M. Leung, "Communications, caching, and computing oriented small cell networks with interference alignment," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 29–35, Sep. 2016.

[51] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.

[52] M. F. Bari *et al.*, "Data center network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 909–928, 2nd Quart., 2013.

[53] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*. London, U.K: Springer, 1995.

[54] D. R. Smith and W. Whitt, "Resource sharing for efficiency in traffic systems," *Bell Syst. Tech. J.*, vol. 60, no. 1, pp. 39–55, Jan. 1981.

[55] C. Y. Yeoh, M. H. Mokhtar, A. A. A. Rahman, and A. K. Samingan, "Performance study of LTE experimental testbed using OpenAirInterface," in *Proc. IEEE Int. Conf. Adv. Commun. Technol.*, Jan./Feb. 2016, pp. 617–622.

[56] N. Mharsi, M. Hadji, D. Niyato, W. Diego, and R. Krishnaswamy, "Scalable and cost-efficient algorithms for baseband unit (BBU) function split placement," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[57] L. A. Adamic and B. A. Huberman, "Zipf's law and the Internet," *Glottometrics*, vol. 3, no. 1, pp. 143–150, 2002.

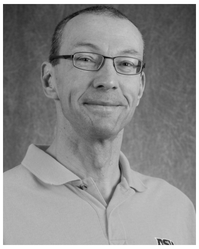
[58] J. Tang, L. Teng, T. Q. S. Quek, T.-H. Chang, and B. Shim, "Exploring the interactions of communication, computing and caching in cloud RAN under two timescale," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–6.

PRATEEK SHANTHARAMA received the B.E. degree in electronics and communication engineering from the Siddaganga Institute of Technology, Tumkur, India, in 2014, and the M.Tech. degree in computer network engineering from The National Institute of Engineering, Mysore, India, in 2016. He is currently pursuing the Ph.D. degree with Arizona State University, Tempe. His current research interests lie in wireless communication, networking, 5G, and SDN.

AKHILESH S. THYAGATURU received the Ph.D. degree in electrical engineering from Arizona State University, Tempe, in 2017. He was with Qualcomm Technologies, Inc., San Diego, CA, USA, as an Engineer, from 2013 to 2015. He is currently an Engineer with the Intel Software and Services Group, Chandler, AZ, USA, and an Adjunct Faculty with the School of Electrical, Computer, and Energy Engineering, Arizona State University. He serves as a reviewer for various journals including the *IEEE Communications Surveys and Tutorials*, the IEEE TRANSACTIONS OF NETWORK AND SERVICE MANAGEMENT, and *Optical Fiber Technology*.

NURULLAH KARAKOC (S'18) received the B.Sc. and M.Sc. degrees in electrical and electronics engineering from Bilkent University, Turkey, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with Arizona State University, Tempe. His current research interests lie in wireless communications, networking, and optimization theory.

LORENZO FERRARI (S'14) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Modena, Italy, in 2012 and 2014, respectively, and the Ph.D. degree in electrical engineering from Arizona State University, Tempe. He is currently a Senior Engineer with Qualcomm Technologies, Inc. His research interest lies in the broad area of wireless communications and signal processing. He has received the IEEE SmartGridComm 2014 Best Student Paper Award for the paper The Pulse Coupled Phasor Measurement Unit.



MARTIN REISSLEIN (S'96–M'98–SM'03–F'14) received the Ph.D. degree in systems engineering from the University of Pennsylvania in 1998. He is currently a Professor with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe. He is Chair of the Steering Committee of the IEEE Transactions on Multimedia. He currently serves as an Associate Editor for the IEEE Transactions on Mobile Computing, the IEEE Transactions on Education, and the IEEE Access, and *Computer Networks*. He is an Associate Editor-in-Chief for the IEEE Communications Surveys and Tutorials and a Co-Editor-in-Chief of *Optical Switching and Networking*.

ANNA SCAGLIONE (F'11) received the M.Sc. degree in 1995 and the Ph.D. degree in 1999. She was an Assistant Professor with the University of New Mexico (2000–2001) and Cornell University (2001–2006), an Associate Professor with Cornell University (2006–2008), and a Professor of electrical engineering with UC at Davis, Davis, (2008–2010), and UC at Davis (2010–2014). She is currently a Professor in electrical and computer engineering with Arizona State University, Tempe. Her research is in modeling, analyzing, and designing mechanisms for reliable and efficient networked systems. Her expertise is in statistical signal processing, wireless communications, and energy delivery systems. She was a recipient of the 2000 IEEE Signal Processing Transactions Best Paper Award, the 2013 IEEE Donald G. Fink Prize Paper Award and, with her student, of the 2013 IEEE Signal Processing Society Young Author Best Paper Award.