HSDRAN: Hierarchical Software-Defined Radio Access Network for Distributed Optimization

Ruozhou Yu, *Student Member, IEEE*, Guoliang Xue, *Fellow, IEEE*, Mehdi Bennis, *Senior Member, IEEE*, Xianfu Chen, *Member, IEEE*, Zhu Han, *Fellow, IEEE*

***Abstract*—The drastic growth of mobile traffic greatly challenges the capacity of mobile infrastructures. Dense deployment of low-power small cells helps alleviate the congestion in the radio access network, yet it also introduces large complexity for network management. Software-defined radio access network has been proposed to tackle the added complexity. However, existing software-defined solutions rely on a fully centralized control plane to make decisions for the whole network, which greatly limits the scalability and responsiveness of the control plane. In this paper, we propose a hierarchical software-defined radio access network architecture. The proposed architecture leverages the hierarchical structure of radio access networks, deploying additional local controllers near the network edge. Utilizing the intrinsic locality in radio access networks, it offloads control tasks from the central controller to local controllers with limited overhead introduced. Under the architecture, a distributed optimization framework is proposed, and a typical optimization problem is studied to illustrate the effectiveness of the proposed architecture and framework. Both analysis and experiments validate that the proposed architecture and framework can improve the network objective during the optimization, meanwhile balancing load and improving scalability and responsiveness.**

***Keywords*—Mobile 5G HetNets, radio access network, software-defined networking, distributed optimization**

I. Introduction

Mobile traffic has undergone drastic growth in the last decade, owing to the advances of wireless broadband technologies and the wide spread of smart devices. Such growth greatly challenges the capacity of the current cellular infrastructure. A major technology invented to tackle this growth is the heterogeneous cellular networks (HetNets), which introduce densely-deployed low-power small base stations (SBSs) to reduce interference and increase system capacity.

The dense deployment of SBSs brings new challenges to cellular radio access networks (RANs). First, large signaling and management overhead has been brought about by the heterogeneous location, channel, power and backhaul characteristics of base stations (BSs). Second, interference management becomes more complex due to more coupled resource allocation among neighboring macro BSs (MBSs) and SBSs. If not properly managed, such interference could greatly impact the throughput of users, especially users served by SBSs.

Software-defined RAN (SDRAN) is a recently proposed concept to tackle these issues [**?**]. SDRAN decouples the control plane and the data plane in the RAN, concentrating control decisions to the control plane. In common SDRAN architectures, a central controller aggregates information from the entire network, and globally makes decisions for every data plane element. This approach avoids the decisional overhead at data plane elements, and offers the opportunity for flexible and coordinated management in the entire RAN.

However, such benefits do not come without a cost. A major concern is the control plane scalability. In a RAN, data plane elements are geographically distributed over a large area. This leads to high backhaul latency from the central controller. On the other hand, the number of end-users in a RAN can be huge. The central controller would incur large computation and communication overhead if per-user decisions are to be made. Also, resource allocation and user management typically require real-time decision making and execution due to the frequent dynamics in RANs. The control plane needs to make timely decisions to ensure fine-grained and in-time control.

To address these issues, we propose a Hierarchical SDRAN (HSDRAN) architecture for 5G HetNets [**?**], which can realize the flexibility and coordination of SDRAN, yet avoiding large centralized load and high latency of a fully centralized control plane. The proposed architecture leverages the hierarchical structure of modern RANs, and divides the control plane into the global controller (GC) and a set of local controllers (LCs). Our insight is to utilize the intrinsic locality in the RAN, *i.e.*, each BS's resource allocation only affects nearby BSs, and each user only receives sufficiently strong signals from nearby BSs. Hence each LC is able to offload many local tasks from the GC with limited coordination incurred. HSDRAN can thus achieve load balancing and scalability, and also improve responsiveness by making decisions at the edge.

To better illustrate how HSDRAN achieves these goals, we propose a distributed optimization framework for HSDRAN. As an illustrative example of implementing distributed optimization in our architecture and framework, we study a typical optimization problem of user association and downlink resource allocation in RANs. The problem jointly optimizes user association and downlink resource allocation, taking into

Yu and Xue ({ruozhouy, xue}@asu.edu) are with Arizona State University, Tempe, AZ 85287. Bennis (bennis@ee.oulu.fi) is with University of Oulu, Oulu 90014, Finland. Chen (xianfu.chen@vtt.fi) is with VTT Technical Research Centre of Finland. Han (zhan2@mail.uh.edu) is with University of Houston, Houston, TX 77004, USA. This research was supported in part by NSF grants 1646607, 1547201, 1456921, 1443917, 1405121, 1457262 and 1461886, and TEKES grants 2364/31/2014 and 2368/31/2014. The information reported here does not reflect the position or the policy of the funding agencies.

account bandwidth, backhaul and power constraints. We solve it using a distributed algorithm. While the proposed problem and algorithm resembles the existing distributed optimization algorithms in the literature [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], we show how to implement this algorithm in our framework, with a properly designed task offloading scheme, feasibility enforcement mechanism, and analysis of their storage and communication overhead at each controller. Our proposed method and analysis can be easily adopted to implement the above mentioned existing algorithms in HSDRAN. Both analysis and experiments show that the proposed architecture and framework achieves load balancing, scalability and responsiveness, and can also gradually improve network objective before convergence.

Our contributions are summarized as follows:

- We propose a novel hierarchical architecture for SDRAN (HSDRAN), which provides flexible and coordinated control with load balancing, scalability and responsiveness.
- Based on HSDRAN, we further propose a distributed optimization framework which features delegation-based implementation of distributed algorithms. We illustrate its benefits through our proposed solution and analysis of a typical network optimization problem in HetNets.
- Both analysis and simulation experiments show that the proposed HSDRAN achieves the desired goals.

The rest of this paper is organized as follows. Section **??** introduces related work. Section **??** presents our proposed HSDRAN architecture. Section **??** presents the optimization problem we study, our primal-dual solution, and its distributed implementation in HSDRAN. Section **??** shows performance evaluation via simulations. Section **??** concludes this paper.

II. Background and Related Work

A. SDRAN and Hierarchical SDN

In the control plane, researches on SDRAN start from SoftRAN [**?**]. SoftRAN abstracts the RAN as a *Big Base Station*, in which radio resources can be coordinated among many BSs. V-Cell [**?**] proposes another abstraction, in view of not the network operator but the user, where multiple BSs controlled by a central controller form a *no-handover zone* in which each user has the illusion that it is connected to only one BS. RadioVisor [**?**] proposes to slice the RAN into multiple slices, enabling multi-operator sharing of the RAN. The authors formulated the multi-operator sharing problem and gave a heuristic solution. [**?**] elaborates architectural insights on how to utilize SDRAN to improve RAN energy efficiency. [**?**] proposes that the combination of SDN and RAN should also incorporate the Cloud-RAN (C-RAN) architecture, which decouples signal processing from signal transmission. [**?**] gives a comprehensive survey on existing SDRAN architectures. All these works consider a fully centralized control plane for SDRAN, which has scalability and responsiveness issues. Recently, [**?**] proposes a two-layer SDRAN architecture that is similar to ours. While it uses local controllers merely to manage Device-to-Device (D2D) communications, we focus on offering generic distributed optimization for SDRANs.

Data plane researches start earlier than control plane. Software-defined radio (SDR) has been proposed for over a decade [**?**], previously to support cognitive radio operations. Recently, its programmability are utilized to support SDRAN. Based on this, recent works propose more advanced technologies for SDRAN data plane. For example, OpenRadio [**?**] proposes a programmable data plane that is tailored for SDRAN. It provides a modular and declarative interface for programming the data plane, including graph-based representations and operator rules. PRAN [**?**] and VHEL [**?**] both propose to virtualize signal processing for BSs, moving it to general-purpose servers, hence enabling its programmability.

Many other researches focus on optimization and algorithm design for SDRAN [**?**], [**?**], [**?**], [**?**]. While they optimize different aspects of SDRAN, they do not address the intrinsic scalability issue and overhead that come with the fully centralized architecture. Other problems studied include security [**?**], edge caching [**?**], [**?**], green networking [**?**], etc.

SDN has been brought into mobile core networks. For example, SoftCell [**?**] proposes an SDN architecture for mobile core networks, which aggregates user flows at user, BS and policy levels to improve scalability. SoftMoW [**?**] proposes a recursive and reconfigurable architecture for mobile core networks, which features network-wide optimization functions including routing, handovers, etc. Since core networks consist of switches and gateways rather than BSs and users, core SDN solutions are mostly in the perspective of flows, which is intrinsically different from in the RAN.

Scalability of SDN control plane in Wireless Local Area Networks (WLANs) has been recently studied. For example, Ali-Ahmad *et al.* [**?**] described using local controllers to control a subset of BSs, and perform local optimizations of several control tasks. Cwalinski *et al.* [**?**] deployed local agents at wireless access points, which selectively forward packets to the global controller. The above proposals mainly focus on offloading certain simple control tasks to local controllers, which can be decided only based on local information. Our proposed HSDRAN architecture can not only offer the same offloading as the above, but also realize distributed optimization in a coordinated manner.

Distributed control plane has also been explored in wired networks. Two types of control distribution schemes have been studied: flat distribution [**?**] and hierarchical distribution [**?**], [**?**], [**?**]. The former deploys multiple controllers working as peers, hence no controller has the global view of the network. The latter has a central controller overseeing all local ones; the central controller maintains a global view, while local controllers utilize locality for task offloading. The difference between our work and the above is that, in wired networks, the impact of network locality is very limited, since each flow can have two arbitrarily far-away end-points; yet in wireless environments, such locality is common and has great impact on network performance, due to the geographical distribution of radio resources and the natural hierarchical structure of RANs.

B. Distributed Optimization in HetNets

Network optimization has been extensively studied in the context of cellular access networks and HetNets. We focus

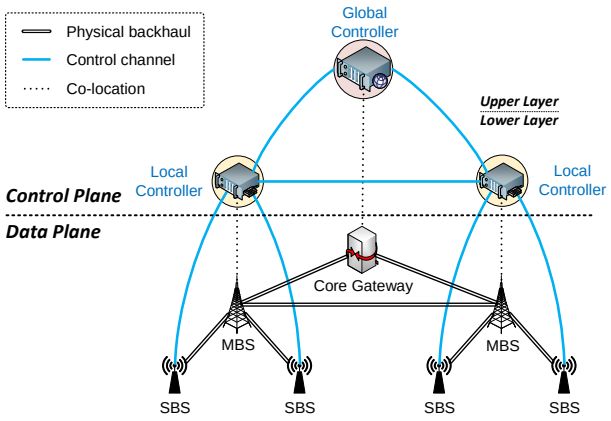


Fig. 1: HSDRAN architecture overview

on distributed optimization for the sake of scalability and load balancing, which has been studied in many works [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. They study network optimization with various variables (user association, user resource allocation, etc.), constraints (bandwidth, backhaul, power, etc.), and objectives (proportional fair, max-min fair, etc.). The optimization problem we study in Section **??** resembles the above, but considers different sets of variables, constraints and objectives. In fact, our proposed optimization framework can easily implement these existing algorithms in HSDRAN, with properly designed task offloading schemes. The novelty of this paper lies in a proposed architecture for overhead-limited distributed optimization, as well as a framework to implement such optimization in the architecture.

III. HSDRAN Architecture

A. Design Goal

We propose an architecture that achieves the following goals:

- **Optimization**: It should automatically optimize network parameters based on global network states. Moreover, it should utilize the intermediate solutions to improve network performance before the optimization converges.
- **Scalability**: It should be able to control a large-scale RAN, which spans the coverage area of multiple MBSs and contains thousands to hundreds of thousands of users.
- **Load balancing**: It should balance the computational and management load among multiple controllers, reducing the load incurred at any single controller. In addition, load balancing must not introduce high communication overhead between controllers.
- **Responsiveness**: It should quickly respond to network dynamics such as user movements, network failures, etc.

B. Architecture

Fig. **??** shows an overview of HSDRAN. Following the principle of SDN, HSDRAN splits the network into the control plane and the data plane. Unlike existing SDRAN architectures, HSDRAN features a two-layer control plane, where the upper layer consists of a globally centralized controller (GC), and the lower layer consists of multiple local controllers (LCs).

Global controller: The GC commonly resides in the network core, e.g., at the Mobility Management Entity (MME) in Long-Term Evolution (LTE). As a central point far away from the network edge, the GC is difficult to aggregate information and make responsive and fine-grained decisions regarding every user. To address this issue, the GC offloads its control tasks to the LCs, including those requiring fast response (e.g., real-time allocation) and those that requires only local information (e.g., intra-MBS handovers). Meanwhile, the GC gathers aggregated information from the LCs, and makes global and coarse-grained decisions (e.g., network-wide power management and long-term load balancing). This effectively reduces the computation, storage and communication overhead at the GC.

An important task for the GC is to initiate network-wide optimization. Based on network information aggregated from LCs, the GC can set different objectives and constraints, and distribute the optimization tasks among all LCs. The GC may also be involved in the optimization if needed. An example of such optimization is shown in Section **??**.

Local controller: The LCs reside near the network edge. In HSDRAN, each LC is co-located with an MBS in the network, and controls the MBS plus all SBSs associated with this MBS. The reason is several-fold. First, an MBS typically has sufficient resources to host a controller. For example, an MBS may be equipped with general-purpose servers that can run control applications, and fiber backhaul links that offer high bandwidth to accommodate inter-controller traffic. Second, an MBS is closer to the network edge, as most SBSs have direct connection to at least one MBS. Moreover, an MBS and associated SBSs typically serve no more than several thousands of users, which is a reasonable number of devices to be controlled by a centralized entity.

Each LC has several types of tasks. First, it makes local decisions regarding the local network. For example, a local MBS–SBS handover does not need to involve the GC, and is handled by the LC. Second, it aggregates information for the GC, and executes high-level instructions from the GC regarding its domain. For example, an LC reports local power consumption to the GC, and gets instructions on power saving if power is over-consumed in the network during the past time period. Last but not least, the LCs cooperatively carry out network-wide optimization based on the instructions of the GC. Each LC stores local variables and constants, conducts local computations, and exchanges results with each other if necessary, as will be shown in Section **??**. Task offloading to LCs not only reduces overhead at the GC and improves scalability, but also improves responsiveness to local dynamics.

Control channel: The control channels carry inter-controller or inter-plane communications, which are logically separated from the data plane network. While deploying physically out-of-band control channels provides the best performance isolation and robustness, it is very cost-inefficient due to the wide geographical distribution of RANs. Hence it is preferred to utilize the existing backhaul links to establish these channels.

Two types of traffic are carried in these channels. The first

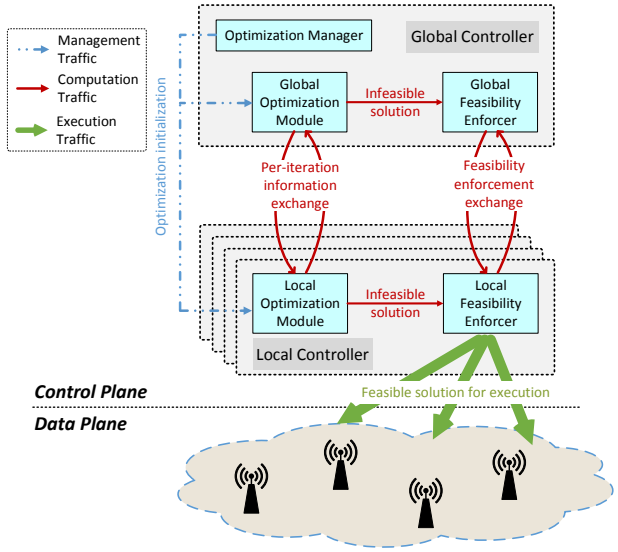


Fig. 2: Distributed optimization in HSDRAN.

type is inter-controller communications (*eastbound traffic*[1]) that coordinate decision making between controllers. This type of traffic is carried on the inter-LC links and the GC–LC links. Since the LCs reside with the MBSs which commonly have high-speed fiber links, these links are sufficient to carry the eastbound traffic. The other type is control primitives and raw data transmitted between LCs and data plane elements. Since SBSs commonly have limited backhaul, control traffic on these links (*southbound traffic*) should be minimal to minimize interference to data plane traffic. Only direct commands and necessary information should be disseminated via these links.

IV. Distributed Optimization in HSDRAN

Fig. **??** shows an overview of the optimization framework in the proposed architecture. The Optimization Manager at the GC initiates and supervises the optimization process. It instructs the Optimization Modules at both GC and LCs for coordinated optimization. The Optimization Modules carry out the distributed optimization. They conduct local computations and exchange results with each other during the optimization.

Many existing optimization algorithms are iterative methods that do not guarantee feasibility until convergence [**?**], [**?**]. The Feasibility Enforcer transforms the intermediate (possibly infeasible) solutions to feasible ones, via additional computation and information exchange. This way, the network can utilize these solutions to gradually improve the network objective without waiting for convergence. No computation happens in the data plane, as only the solutions are translated into control primitives and sent to the data plane for execution.

[1]Note that while the first type of traffic involves upward traffic from LCs to GCs in the proposed controller hierarchy, we still refer to such traffic as eastbound traffic. The term *northbound* commonly describes the interface between the control plane and the management plane in the SDN convention.

To illustrate the benefits of our architecture and framework, we study a concrete optimization problem in the RAN, and implement a classic algorithm of the problem in our architecture and framework. We study joint user association and downlink resource allocation in HetNet RANs, considering radio, backhaul and power resource constraints. This is a typical network optimization problem as studied in many works [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], with a different set of constraints and objective from the above. Implementation of our proposed algorithm (and other similar algorithms in aforementioned researches) in fully centralized SDRANs results in low scalability, low responsiveness, and large overhead. On the contrary, our proposed architecture and framework well address these issues via delegation-based task offloading.

A. System Model

The network consists of multiple MBSs, denoted by $\mathcal{M} = \{M_1, \ldots, M_m\}$. Each MBS $M \in \mathcal{M}$ has multiple SBSs, denoted by $\mathcal{S}_M = \{S_1^M, \ldots, S_{n_M}^M\}$. All SBSs are denoted by $\mathcal{S} = \bigcup_{M\in\mathcal{M}} \mathcal{S}_M$. All BSs are denoted by $\mathcal{B} = \mathcal{M} \cup \mathcal{S}$.

We consider several kinds of resources in the network. Radio resources are defined in continuous time-frequency slots for each BS, and can be shared among BSs with frequency reuse. We assume that all BSs share the same set of frequency bands, and define $A > 0$ as the total radio bandwidth of all bands that are shared among all BSs (in unit time). Each BS $B \in \mathcal{B}$ has backhaul capacity of $\beta_B > 0$ for serving user traffic. Further, each BS $B \in \mathcal{B}$ has a renewable power source, which can constantly provide $\rho_B^n \geq 0$ power. Each BS is also connected to the power grid, which will provide energy if the renewable energy is not sufficient. However, the grid power is not free-to-use compared to local renewable power, hence an on-grid power bound $\mathcal{P} \geq 0$ is enforced network-wide to limit the total on-grid power used by all BSs. A BS's power consumption has two parts. The fixed power consumption of BS $B \in \mathcal{B}$ is $\rho_B^f \geq 0$. The dynamic power consumption is defined by power slope $\rho_B^t > 0$, which is the power consumed for transmitting unit bandwidth of radio signal with fixed transmit power. The power consumption of BS B on the grid is thus defined as

$$\rho_B = \left[\rho_B^f + \rho_B^t \cdot a_B - \rho_B^n\right]^+, \tag{1}$$

where $[\cdot]^+$ denotes the projection onto non-negative real numbers, and a_B is the total bandwidth used by BS B. The network-wide power bound is expressed as follows

$$\sum\nolimits_{B\in\mathcal{B}} \rho_B \leq \mathcal{P}. \tag{2}$$

We assume that $\mathcal{P} > \sum_{B\in\mathcal{B}} [\rho_B^f - \rho_B^n]^+$, meaning that the on-grid power is sufficient to cover some radio transmission in addition to all fixed part consumptions of BSs.

A set of users exist in the network, denoted by $\mathcal{U} = \{U_1, \ldots, U_K\}$. Each user receives signals from several BSs, including both MBS and SBS. To avoid strong interference from MBS on users served by SBS, modern HetNets employs almost blank subframes (ABSs), where some subframes are left blank by the MBS for transmissions from/to SBSs. Denote

$P_{B,U}$ as the power received from BS B at user U. For each user U and BS B, the signal-to-interference-and-noise ratio (SINR) during non-ABS is

$$\text{SINR}^{+}_{B,U} = \frac{P_{B,U}}{\sum_{B' \neq B} P_{B',U} + N_0}, \tag{3}$$

and for user U and SBS $S \in \mathcal{S}$, the SINR during ABS is

$$\text{SINR}^{-}_{S,U} = \frac{P_{S,U}}{\sum_{\substack{B' \neq S \\ B' \neq M(S)}} P_{B',U} + N_0}, \tag{4}$$

where $M(B)$ is the MBS with which SBS B is associated, or the MBS itself if B is an MBS, and N_0 is the noise power. The SINR of an MBS during ABS is always 0 for any user.

The spectral efficiency is defined based on the Shannon capacity, where during ABS, it is

$$\eta^{+}_{B,U} = \log_2(1 + \text{SINR}^{+}_{B,U}), \tag{5}$$

and during non-ABS, it is

$$\eta^{-}_{B,U} = \log_2(1 + \text{SINR}^{-}_{B,U}). \tag{6}$$

For each user, only signals with SINR above a threshold Υ can be successfully decoded. We denote $\mathcal{B}_U$ as the set of BSs (namely *candidate BSs*) whose SINR at user U is at least Υ during either ABS or non-ABS, and $\mathcal{U}_B$ as the set of users (namely *candidate users*) whose SINR from B is at least Υ.

TABLE I: Notations

Symbol	Meaning
$\mathcal{M}$	Set of MBSs; $\lvert\mathcal{M}\rvert = m$
$\mathcal{S}_M$	Set of SBSs associated with MBS M; $\lvert\mathcal{S}_M\rvert = n_M$
$\mathcal{S}, \mathcal{B}$	Set of all SBSs and all BSs
$\mathcal{U}$	Set of all users; $\lvert\mathcal{U}\rvert = K$
A	Total radio bandwidth shared by all BSs
β_B	Backhaul capacity of BS B
$\mathcal{P}$	Global on-grid power consumption limit
ρ^n_B	Renewable power at BS B
ρ^f_B	Fixed power consumption at BS B
ρ^t_B	Dynamic power slope for transmission at BS B
ρ_B	On-grid power consumption of BS B
$\eta^{+}_{B,U}, \eta^{-}_{B,U}$	Spectral efficiency during non-ABS and ABS
Υ	Global SINR threshold
$\mathcal{B}_U$	Set of BSs with above-threshold SINR at user U
$\mathcal{U}_B$	Set of users with above-threshold SINR at BS B
r_U	Aggregate bandwidth of user U (variable)
$x_{B,U}$	Fraction of user U served by BS B (variable)
$a^{+}_{B,U}, a^{-}_{B,U}$	Bandwidth allocation during non-ABS and ABS (variables)
α_M	ABS ratio at MBS M (variable)

Table **??** summarizes the notations used in this section.

B. Problem Formulation

We study joint user association and downlink resource allocation in the RAN. In user association, we decide the serving BS(s) for each user. We assume that each user can be associated with multiple BSs, and define variable $x_{B,U} \in [0, 1]$ as the fraction of user U served by BS B. In practice, $x_{B,U}$ can be interpreted as the long-term association of the user, who may be switched among multiple BSs for load balancing [**?**].

In resource allocation, we allocate resources in two perspectives. First, we need to decide the fraction of radio resources that are dedicated to ABSs for each MBS, and we use variable $\alpha_M \in [0, 1]$ to denote this fraction for each MBS $M \in \mathcal{M}$. Second, we need to allocate radio resources, for both ABSs and non-ABSs, to each user, based on their associations. We use variable $a^{+}_{B,U} \in [0, A]$ to denote the bandwidth allocated for user U at BS B during non-ABS, and $a^{-}_{B,U} \in [0, A]$ to denote the bandwidth allocated during ABS. Since MBSs cannot transmit during ABSs, we have $a^{-}_{M,U} = 0$ for any MBS M and user U. In practice, radio resources are commonly sliced into unit-length time slots and subcarriers. However, variables α_M, $a^{+}_{B,U}$ and $a^{-}_{B,U}$ take real numbers, which can be interpreted as the long-term allocation for SBSs and users respectively. Finally, we use variable $r_U \geq 0$ to denote the aggregate rate of user U from all candidate BSs.

The problem we study is formulated as follows:

$$\max \quad \sum\nolimits_{U \in \mathcal{U}} w_U \log(r_U) \tag{7}$$

$$\text{s.t.} \quad r_U \leq \sum_{B \in \mathcal{B}_U} (\eta^{+}_{B,U} a^{+}_{B,U} + \eta^{-}_{B,U} a^{-}_{B,U}), \quad \forall U \in \mathcal{U}; \tag{8}$$

$$\sum\nolimits_{B \in \mathcal{B}_U} x_{B,U} = 1, \quad \forall U \in \mathcal{U}; \tag{9}$$

$$a^{+}_{B,U} + a^{-}_{B,U} \leq A \cdot x_{B,U}, \quad \forall U \in \mathcal{U}, B \in \mathcal{B}_U; \tag{10}$$

$$a^{-}_{M,U} = 0, \quad \forall M \in \mathcal{M}, U \in \mathcal{U}_M; \tag{11}$$

$$\sum\nolimits_{U \in \mathcal{U}_B} a^{-}_{B,U} \leq A \cdot \omega_B \cdot \alpha_{M(B)}, \quad \forall B \in \mathcal{B}; \tag{12}$$

$$\sum\nolimits_{U \in \mathcal{U}_B} a^{+}_{B,U} \leq A \cdot (1 - \alpha_{M(B)}), \quad \forall B \in \mathcal{B}; \tag{13}$$

$$\sum_{U \in \mathcal{U}_B} (\eta^{+}_{B,U} a^{+}_{B,U} + \eta^{-}_{B,U} a^{-}_{B,U}) \leq \beta_B, \forall B \in \mathcal{B}; \tag{14}$$

$$\sum_{B \in \mathcal{B}} \left[\rho^f_B + \rho^t_B \cdot \sum_{U \in \mathcal{U}_B} (a^{+}_{B,U} + a^{-}_{B,U}) - \rho^n_B \right]^{+} \leq \mathcal{P}; \tag{15}$$

$$x_{B,U}, \alpha_M \in [0, 1], a^{+}_{B,U}, a^{-}_{B,U} \in [0, A], r_U \in [0, maxSE \cdot A]. \tag{16}$$

Explanation: The objective function (**??**) is the weighted proportional fairness of user rates, where w_U is the weight of user U. In the objective function, user weights are determined by the actual types of traffic of the users; for example, real-time video traffic should have larger weights than static webpage inquiries. These weights are periodically updated by each user's local controller based on the traffic pattern of each user in the past period. Constraint (**??**) defines the user rate bounded by the allocated radio resources (variables) and the spectral efficiencies (constants) from all candidate BSs. Constraint (**??**) states the fractional constraint of each user. Constraint (**??**) bounds the allocated resources by the user association, enforcing that the resources allocated from BS B cannot exceed the fraction served by this BS. Constraint (**??**) states the non-transmission rule for MBS during ABS. Constraint (**??**) bounds the total amount of ABS resources allocated to users by the allocated ABS resources from the MBS, where ω_B is an indicator of whether BS B is an MBS ($\omega_B = 0$) or an SBS ($\omega_B = 1$).

Constraint (**??**) similarly bounds the non-ABS resources to users. Constraint (**??**) enforces the backhaul bound of the aggregate user rate at each BS. Constraint (**??**) enforces the network-wide on-grid power bound. Constraint (**??**) specifies the range of each variable, where $maxSE$ is the maximum possible spectral efficiency.

For the above problem, the Slater's condition is satisfied since $\mathcal{P} > \sum_{B\in\mathcal{B}} [\rho_B^f - \rho_B^n]^+$, $\beta_B > 0$ for $\forall B \in \mathcal{B}$, $A > 0$, and $\eta_{B,U}^+ > 0$ and $\eta_{B,U}^- > 0$ for $\forall B \in \mathcal{B}, \forall U \in \mathcal{U}_B$. To see this, note that we can always set $x_{B,U} = 1/|\mathcal{B}_U|$ for $U \in \mathcal{U}$ and $B \in \mathcal{B}_U$, and $\alpha_M = 0.5$ for $M \in \mathcal{M}$. Resource allocations $a_{B,U}^+$ and $a_{B,U}^-$ can be made to strictly satisfy Constraints (**??**), (**??**), (**??**), (**??**), and (**??**), by assigning an arbitrarily small value $\epsilon > 0$ to $a_{B,U}^+$ and $a_{B,U}^-$ for any $B \in \mathcal{B}$ and $U \in \mathcal{U}_B$, except $a_{B,U}^- = 0$ for MBS $B \in \mathcal{M}$. Also, user rates r_U can be set as strictly smaller than the sum of rates from all candidate BSs in Constraint (**??**). Since the problem is maximizing a concave function subject to convex constraints, the Slater's condition ensures strong duality, meaning that the primal optimal objective value equals the dual optimal objective value. Hence we can use primal-dual algorithms like the dual subgradient method to solve it.

C. Dual Subgradient Method

While there are many methods to solve this problem, we solve it using the *dual subgradient method* (DSM) following existing work [**?**], [**?**]. DSM is preferred in large-scale optimization due to its possibility for distributed realization via dual decomposition, hence it will benefit from the above proposed HSDRAN architecture. Note that primal and dual decompositions are widely employed in wireless optimizations, which will all benefit from HSDRAN.

Dual problem: Define $\boldsymbol{z} = (\boldsymbol{r}, \boldsymbol{x}, \boldsymbol{a}^+, \boldsymbol{a}^-, \boldsymbol{\alpha})$ as the primal variable vector, where **bold symbols** denote the corresponding *variable vectors*. To start with, we define the subspace for primal variables as

$$\Pi = \{\boldsymbol{z} \geq 0 \,|\, (\textbf{??}),\ (\textbf{??}),\ (\textbf{??})\}. \tag{17}$$

For the other constraints, we associate dual variables γ_U for (**??**), $\sigma_{B,U}$ for (**??**), μ_B for (**??**), ν_B for (**??**), λ_B for (**??**), and δ for (**??**). Define $\boldsymbol{p} = (\boldsymbol{\gamma}, \boldsymbol{\sigma}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\delta})$ as the dual variable vector, the dual subspace is defined as $\{\boldsymbol{p} \geq 0\}$.

The Lagrangian of the primal problem is as follows

$$\begin{aligned}
\mathcal{L}(\boldsymbol{z}, \boldsymbol{p}) &= \textstyle\sum_{U\in\mathcal{U}} w_U \log(r_U) - \boldsymbol{p}^T \boldsymbol{g}(\boldsymbol{z}) \\
&= \textstyle\sum_{U\in\mathcal{U}} w_U \log(r_U) \\
&- \textstyle\sum_{U\in\mathcal{U}} \gamma_U \left(r_U - \sum_{B\in\mathcal{B}_U} \left(\eta_{B,U}^+ a_{B,U}^+ + \eta_{B,U}^- a_{B,U}^- \right) \right) \\
&- \textstyle\sum_{U\in\mathcal{U}} \sum_{B\in\mathcal{B}_U} \sigma_{B,U} \left(a_{B,U}^+ + a_{B,U}^- - A \cdot x_{B,U} \right) \\
&- \textstyle\sum_{B\in\mathcal{B}} \mu_B \left(\sum_{U\in\mathcal{U}_B} a_{B,U}^- - A \cdot \omega_B \cdot \alpha_{M(B)} \right) \\
&- \textstyle\sum_{B\in\mathcal{B}} \nu_B \left(\sum_{U\in\mathcal{U}_B} a_{B,U}^+ - A \cdot (1 - \alpha_{M(B)}) \right) \\
&- \textstyle\sum_{B\in\mathcal{B}} \lambda_B \left(\sum_{U\in\mathcal{U}_B} (\eta_{B,U}^+ a_{B,U}^+ + \eta_{B,U}^- a_{B,U}^-) - \beta_B \right)
\end{aligned} \tag{18}$$

$$- \delta \left(\sum_{B\in\mathcal{B}} \left[\rho_B^f + \rho_B^t \cdot \sum_{U\in\mathcal{U}_B} (a_{B,U}^+ + a_{B,U}^-) - \rho_B^n \right]^+ - \mathcal{P} \right),$$

where $\boldsymbol{g}(\cdot)$ denotes the vector of all primal constraint functions except Constraints (**??**), (**??**) and (**??**). The dual problem is

$$\min_{\boldsymbol{p}\geq 0} \mathcal{D}(\boldsymbol{p}), \tag{19}$$

where

$$\mathcal{D}(\boldsymbol{p}) = \max_{\boldsymbol{z}\in\Pi} \mathcal{L}(\boldsymbol{z}, \boldsymbol{p}). \tag{20}$$

Given the dual problem, for any $\boldsymbol{p} \geq 0$, the subgradients of $\mathcal{D}(\boldsymbol{p})$ consists of $\boldsymbol{g}(\boldsymbol{z})$ for any $\boldsymbol{z} \in \Pi$ such that $\mathcal{L}(\boldsymbol{z}, \boldsymbol{p}) = \mathcal{D}(\boldsymbol{p})$, as observed in [**?**]. In other words, the set of subgradients of the dual function (**??**) is given by the set of primal constraint function values at any primal solution in Π that achieves the maximum of the dual function. Hence we can utilize this simple subgradient structure in our DSM.

Dual subgradient method: DSM is an iterative method that updates the primal and dual solutions in each iteration, until convergence. Initially, the primal variables are initialized to some value in Π, and the dual variables are initialized to 0. We can always use an all-0 solution as the initial primal point, though other initial points, such as evenly distributed user association and resource allocation among all entities, will also work with little impact on performance. Then, in each iteration i, the algorithm conducts the following two steps:

Primal update: The primal variables in the i-th iteration (denoted by $\boldsymbol{z}^{(i)}$) are obtained by solving the following problem

$$\boldsymbol{z}^{(i)} = \arg\max_{z\in\Pi} \mathcal{L}(\boldsymbol{z}, \boldsymbol{p}^{(i-1)}). \tag{21}$$

Dual update: The dual variables in the i-th iteration (denoted by $\boldsymbol{p}^{(i)}$) are updated as

$$\boldsymbol{p}^{(i)} = [\boldsymbol{p}^{(i-1)} + \theta_i \boldsymbol{g}(\boldsymbol{z}^{(i)})]^+, \tag{22}$$

where θ_i is the step size for the i-th iteration.

Remark 4.1: Based on classic convergence results of the subgradient method, DSM converges to within a given error bound of the optimal solution when constant step size or step length is used, and to the optimal when diminishing step sizes are used [**?**]. This is not affected by the specific implementation of the algorithm, for example, the decomposition-based implementation proposed below.

D. Distributed Algorithm via Dual Decomposition

DSM features dual decomposition-based distributed implementation of each iteration. In particular, each primal update or dual update step is decomposed into per-user update, per-BS update, and per user-BS pair update.

1) Primal Update Decomposition: Recall that the primal update step is to obtain the primal variable values that maximize the Lagrangian function in (**??**). Define $\boldsymbol{x}_U = (x_{B,U} \,|\, B \in \mathcal{B}_U)$ to be the vector of association variables for user U, and $\boldsymbol{a}_B = (a_{B,U}^+, a_{B,U}^- \,|\, U \in \mathcal{U}_B)$ to be the vector of all user allocation variables associated with BS B, for both ABSs and non-ABSs. Also define $R = [0, maxSE \cdot A]$, $X_U = \{\boldsymbol{x}_U \in [0,1]^{|\boldsymbol{x}_U|} \,|\, \sum_{B\in\mathcal{B}_U} x_{B,U} = 1\}$, $A_B =$

$$\max_{z\in\Pi}\mathcal{L}(\boldsymbol{z},\boldsymbol{p}(i-1)) = \sum\nolimits_{U\in\mathcal{U}}\max_{r_U\in R}\left(w_U\log(r_U)-\gamma_U^{(i-1)}r_U\right)+\sum\nolimits_{U\in\mathcal{U}}A\max_{\boldsymbol{x}_U\in X_U}\sum\nolimits_{B\in\mathcal{B}_U}\sigma_{B,U}^{(i-1)}x_{B,U}$$
$$+A\sum_{M\in\mathcal{M}}\max_{\alpha_M\in\Lambda}\alpha_M\sum_{B\in\mathcal{S}_M\cup\{M\}}(\mu_B^{(i-1)}\cdot\omega_B-\nu_B^{(i-1)})+\sum_{B\in\mathcal{B}}\max_{\boldsymbol{a}_B\in A_B}\left(\sum_{U\in\mathcal{U}_B}\left(\zeta_{B,U}^{-,(i-1)}a_{B,U}^{-}+\zeta_{B,U}^{+,(i-1)}a_{B,U}^{+}\right)\right. \quad (23)$$
$$\left.-\delta^{(i-1)}\left[\rho_B^f+\rho_B^t\cdot\sum\nolimits_{U\in\mathcal{U}_B}(a_{B,U}^{+}+a_{B,U}^{-})-\rho_B^n\right]^+\right)+A|\mathcal{B}|+\sum\nolimits_{B\in\mathcal{B}}\lambda_B^{(i-1)}\beta_B+\delta^{(i-1)}P$$

$\{\boldsymbol{a}_B\in[0,A]^{|\boldsymbol{a}_B|}\,|\,\text{if } B\in\mathcal{M},\, a_{B,U}^{-}=0 \text{ for } \forall U\in\mathcal{U}_B\}$, and $\Lambda=[0,1]$ to be the projection of subspace Π over each variable or variable vector r_U per $U\in\mathcal{U}$, $\boldsymbol{x}_U$ per $U\in\mathcal{U}$, $\boldsymbol{a}_B$ per $B\in\mathcal{B}$, and α_M per $M\in\mathcal{M}$, respectively. The maximization problem can be re-written as in (**??**), where

$$\zeta_{B,U}^{-,(i-1)}=(\gamma_U^{(i-1)}-\lambda_B^{(i-1)})\eta_{B,U}^{-}-\sigma_{B,U}^{(i-1)}-\mu_B^{(i-1)}, \quad (24)$$
$$\zeta_{B,U}^{+,(i-1)}=(\gamma_U^{(i-1)}-\lambda_B^{(i-1)})\eta_{B,U}^{+}-\sigma_{B,U}^{(i-1)}-\nu_B^{(i-1)}. \quad (25)$$

Based on (**??**), the primal update step at iteration i can be divided into several steps:

1) r_U *update:* For each user U, its rate r_U is updated as $r_U^{(i)}=\min\{\frac{w_U}{\gamma_U^{(i-1)}}, A\cdot maxSE\}$.
2) $x_{B,U}$ *update:* Each user U picks the BS $B^*=\arg\max_B\{\sigma_{B,U}^{(i-1)}\}$, and updates $x_{B^*,U}^{(i)}=1$ and $x_{B',U}^{(i)}=0$ for $B'\in\mathcal{B}_U\setminus\{B^*\}$.
3) α_M *update:* For each MBS M, if $\sum_{B\in\mathcal{S}_M\cup\{M\}}(\mu_B^{(i-1)}\cdot\omega_B-\nu_B^{(i-1)})>0$, $\alpha_M^{(i)}=1$; otherwise, $\alpha_M^{(i)}=0$.
4) $a_{B,U}^{+}, a_{B,U}^{-}$ *update:* User resource allocation is by solving the following optimization problem for each BS $B\in\mathcal{B}$:

$$\max_{\boldsymbol{a}_B\in A_B}\left(\sum\nolimits_{U\in\mathcal{U}_B}\left(\zeta_{B,U}^{-,(i-1)}a_{B,U}^{-}+\zeta_{B,U}^{+,(i-1)}a_{B,U}^{+}\right)\right.$$
$$\left.-\delta^{(i-1)}\left[\rho_B^f+\rho_B^t\cdot\sum\nolimits_{U\in\mathcal{U}_B}(a_{B,U}^{+}+a_{B,U}^{-})-\rho_B^n\right]^+\right). \quad (26)$$

Note that although this involves maximization of a non-linear function, the following greedy algorithm achieves such maximization. For each BS B and each user $U\in\mathcal{U}_B$, both $a_{B,U}^{+,(i)}$ and $a_{B,U}^{-,(i)}$ are initialized to 0. First, we find all users $U\in\mathcal{U}_B$ such that $\zeta_{B,U}^{-,(i-1)}\geq\delta^{(i-1)}\rho_B^t$, and assign $a_{B,U}^{-,(i)}=A$. Next, we find all users $U\in\mathcal{U}_B$ such that $\zeta_{B,U}^{+,(i-1)}\geq\delta^{(i-1)}\rho_B^t$, and assign $a_{B,U}^{+,(i)}=A$. Third, for the rest unassigned variables in $\boldsymbol{a}_B$, we do the following steps: let $a_B^*=(\rho_B^n-\rho_B^f)/\rho_B^t$ be the radio resources that can be served from the renewable power source; if $a_B^*>0$, we then consecutively pick user $U^-=\arg\max_{U\in\mathcal{U}_B}\{\zeta_{B,U}^{-,(i-1)}\geq 0\,|\,a_{B,U}^{-,(i)}=0\}$, and user $U^+=\arg\max_{U\in\mathcal{U}_B}\{\zeta_{B,U}^{+,(i-1)}\geq 0\,|\,a_{B,U}^{+,(i)}=0\}$ (for MBSs, only U^+ is considered); if $\zeta_{B,U^-}^{-,(i-1)}>\zeta_{B,U^+}^{+,(i-1)}$ and $\zeta_{B,U^-}^{-,(i-1)}\geq 0$, we let $a_{B,U^-}^{-,(i)}=\min\{A,a_B^*\}$, and let $a_B^*=a_B^*-a_{B,U^-}^{-,(i)}$; else if $\zeta_{B,U^-}^{-,(i-1)}\leq\zeta_{B,U^+}^{+,(i-1)}$ and $\zeta_{B,U^+}^{+,(i-1)}\geq 0$, we let $a_{B,U^+}^{+,(i)}=\min\{A,a_B^*\}$, and let $a_B^*=a_B^*-a_{B,U^+}^{+,(i)}$; continue this until $a_B^*=0$ or no such user can be found.

2) Dual Update Decomposition: The dual update step can also be divided into several steps based on (**??**): each user U updates γ_U, each BS updates μ_B,ν_B,λ_B, each user–BS pair updates $\sigma_{B,U}$, and globally the network updates δ. Detailed steps for the i-th iteration are shown in (**??**)–(**??**). Note that γ_U is updated at the user side, μ_B,ν_B,λ_B are updated at the BS side, and δ is updated jointly by all BSs. $\sigma_{B,U}$ update involves both BS and user, hence can be conducted at either side based on availability of computational and storage resources.

E. Delegation-based Implementation in HSDRAN

The above distributed algorithm assumes that all BSs, all users and the network core would be involved in the computation process. In practice, such cooperation is subject to control capability and overhead constraints, and thus is not commonly available in modern RANs. For example, user devices are typically out of the control of the network operator; while SBSs are controlled, they may lack either computational resources (Remote Radio Heads in the C-RAN architecture) or network bandwidth (wireless-backhauled SBSs). In HSDRAN, we utilize only the control plane to carry out the computations, hence avoiding overhead in the data plane, meanwhile achieving load balancing compared to fully centralized SDRAN proposals. We use a delegation-based scheme to achieve this goal.

In our proposed scheme, *task delegation* is to find a delegate LC for each user and each BS, which essentially stores data, carries out computation, and exchanges information on behalf of the delegated user or BS. In particular, each MBS's tasks are delegated to its associated LC. Each user is delegated to the LC at the nearest MBS. We use $\mathcal{U}_M^D$ to denote the set of users delegated at the LC at MBS M. Similarly, each SBS delegates all its tasks to the LC at the MBS that it is associated with. Each LC stores the corresponding information and optimization variables for its delegated users and BSs. Information exchange happens only within the control plane, between different LCs and between LCs and the GC. This creates a logical separation between data plane user traffic and control plane management traffic, which is beneficial for performance isolation and guarantee.

Next we analyze the detailed storage and computation delegation process, and its storage and communication overhead. **Storage:** The GC stores the network-wide dual variable δ, and the network-wide power bound $\mathcal{P}$. For each variable involving only the user (r_U and γ_U), it is stored at the delegate LC of the user. For each variable involving only the BS (α_M, μ_B,

$$\gamma_U^{(i)} = \left[\gamma_U^{(i-1)} + \theta_i \left(r_U^{(i)} - \sum\nolimits_{B\in\mathcal{B}_U} \left(\eta_{B,U}^{+} a_{B,U}^{+,(i)} + \eta_{B,U}^{-} a_{B,U}^{-,(i)}\right)\right)\right]^{+} \tag{27}$$

$$\sigma_{B,U}^{(i)} = \left[\sigma_{B,U}^{(i-1)} + \theta_i \left(a_{B,U}^{+,(i)} + a_{B,U}^{-,(i)} - A \cdot x_{B,U}^{(i)}\right)\right]^{+} \tag{28}$$

$$\mu_B^{(i)} = \left[\mu_B^{(i-1)} + \theta_i \left(\sum\nolimits_{U\in\mathcal{U}_B} a_{B,U}^{-,(i)} - A \cdot \omega_B \cdot \alpha_{M(B)}^{(i)}\right)\right]^{+} \tag{29}$$

$$\nu_B^{(i)} = \left[\nu_B^{(i-1)} + \theta_i \left(\sum\nolimits_{U\in\mathcal{U}_B} a_{B,U}^{+,(i)} - A \cdot (1 - \alpha_{M(B)}^{(i)})\right)\right]^{+} \tag{30}$$

$$\lambda_B^{(i)} = \left[\lambda_B^{(i-1)} + \theta_i \left(\sum\nolimits_{U\in\mathcal{U}_B} (\eta_{B,U}^{+} a_{B,U}^{+,(i)} + \eta_{B,U}^{-} a_{B,U}^{-,(i)}) - \beta_B\right)\right]^{+} \tag{31}$$

$$\delta^{(i)} = \left[\delta^{(i-1)} + \theta_i \left(\sum\nolimits_{B\in\mathcal{B}} \left[\rho_B^f + \rho_B^t \cdot \sum\nolimits_{U\in\mathcal{U}_B} (a_{B,U}^{+} + a_{B,U}^{-}) - \rho_B^n\right]^{+} - \mathcal{P}\right)\right]^{+} \tag{32}$$

ν_B and λ_B), it is stored at the delegate LC of the BS. Other variables ($x_{B,U}$, $a_{B,U}^{+}$, $a_{B,U}^{-}$ and $\sigma_{B,U}$) are related to both BS and user, which may have different delegate LCs. In the above primal update step, since each user picks only one BS in each iteration, $x_{B,U}$ should be stored at the user delegate for consistent selection. $a_{B,U}^{+}$ and $a_{B,U}^{-}$ are local resource allocations at each BS, hence are stored at the BS delegate to facilitate local update of all related dual variables including μ_B, ν_B, λ_B. We also let $\sigma_{B,U}$ be stored at the BS delegate, merely to facilitate its local update.

Non-variable storage involves user weights w_U (stored at user delegate), BS–user spectral efficiencies $\eta_{B,U}^{+}$ and $\eta_{B,U}^{-}$ (stored at BS delegate where they are extracted), backhaul capacities β_B (stored at BS delegate), and power parameters $\rho_B^f, \rho_B^n, \rho_B^t$ (stored at BS delegate). Other constants are shared globally, including total bandwidth A and maximum spectral efficiency $maxSE$. To sum up, each LC at MBS M stores $3|\mathcal{S}_M| + 4 + 3\sum_{B\in\mathcal{S}_M\cup\{M\}} |\mathcal{U}_B|$ local variables and $4|\mathcal{S}_M| + 4 + 2\sum_{B\in\mathcal{S}_M\cup\{M\}} |\mathcal{U}_B|$ local constants for delegated BSs, and $2|\mathcal{U}_M^D| + \sum_{U\in\mathcal{U}_M^D} |\mathcal{B}_U|$ local variables and $|\mathcal{U}_M^D|$ local constants for delegated users. The GC stores only one variable, one constant power bound, and other shared information.

Communications: Communications are incurred when information needed for a variable update is stored at other LCs or the GC. Note that most information is stored locally, hence only variables and constants involving both BS and user need to be exchanged, and it only happens when a candidate BS of a user has a different delegate from the user's delegate. We analyze the number of messages (one message carries one constant or variable value) needed in each iteration:

1) $r_U, \alpha_M, \mu_B, \nu_B, \lambda_B$ update: all information is locally stored at delegate LCs, hence no communications needed.
2) $x_{B,U}$ update: user delegate of U needs $\sigma_{B,U}^{(i-1)}$ from the BS delegate if BS $B \in \mathcal{B}_U$ has a different delegate from user U. Since $x_{B,U}$ update is to pick one BS with maximum $\sigma_{B,U}^{(i-1)}$, each LC only needs to transmit the largest $\sigma_{B,U}^{(i-1)}$ value of all its delegated candidate BSs of user U, if any.
3) $a_{B,U}^{+}, a_{B,U}^{-}$ update: variables $\lambda_B^{(i-1)}$, $\sigma_{B,U}^{(i-1)}$, $\mu_B^{(i-1)}$, $\nu_B^{(i-1)}$ and other power constants are local at BS delegate of B. Delegate of B needs $\gamma_U^{(i-1)}$ for each candidate user U, from the user delegate of U if different from delegate of B. Variable $\delta^{(i-1)}$ is needed from the GC.
4) γ_U update: user delegate of U needs the per-user rate $\left(\eta_{B,U}^{+} a_{B,U}^{+,(i)} + \eta_{B,U}^{-} a_{B,U}^{-,(i)}\right)$ from each candidate BS's delegate. Since the update only needs the sum rate, each LC at MBS M transmits the aggregated rate $\sum_{B\in\mathcal{S}_M\cup\{M\}} \left(\eta_{B,U}^{+} a_{B,U}^{+,(i)} + \eta_{B,U}^{-} a_{B,U}^{-,(i)}\right)$ of all candidate BSs of U within its control, to delegate of U.
5) $\sigma_{B,U}$ update: BS delegate of B needs $x_{B,U}^{(i)}$ from the user delegate of U. Note that in the x update, only one BS is selected per-iteration for each user. Hence only the selected BS needs to be aware of the selection, and each LC sends 1 message per delegated user.
6) δ update: the GC only needs the aggregate power consumption from each LC, hence each LC sends 1 message.

The summarized storage and communication overheads can be found in Table **??** and **??** respectively.

TABLE II: Storage Overhead

	LC	GC
Consts	$4\|\mathcal{S}_M\| + 4 + 2\sum_{B\in\mathcal{S}_M\cup\{M\}} \|\mathcal{U}_B\|$	$\|\mathcal{U}_M^D\|$
Vars	$3\|\mathcal{S}_M\| + 4 + 3\sum_{B\in\mathcal{S}_M\cup\{M\}} \|\mathcal{U}_B\|$	$2\|\mathcal{U}_M^D\| + \sum_{U\in\mathcal{U}_M^D} \|\mathcal{B}_U\|$
Vars-PFT	$\sum_{B\in\mathcal{S}_M\cup\{M\}} \|\mathcal{U}_B\|$	0

TABLE III: Average Communication Overhead

	LC	GC
Primal update	$4\|\mathcal{U}_M^D\| + 1$	$\|\mathcal{M}\|$
Dual update	$4\|\mathcal{U}_M^D\| + 1$	$\|\mathcal{M}\|$
PFT	2	$2\|\mathcal{M}\|$

Note that most of the communications are between LCs at nearby MBSs, as users delegated at the LC with one MBS are very unlikely to receive strong-enough signals from other far-away MBSs or SBSs. The GC has very little storage and communication overhead, as it does not need to store per-user and per-BS information. Each LC only stores information regarding local users and BSs, and exchanges information with only nearby LCs. Therefore our proposed method well utilizes locality in RANs to reduce overhead.

Remark 4.2: In the above process, the delegation of users is with no regard to the specific association of users, since

each user–BS pair with SINR above the threshold Υ needs to participate in the optimization process, regardless of the final association of users. The same holds for any optimization that involves user association as variables, *e.g.*, even for user association with massive Multi-Input Multi-Output (MIMO) networks [**?**]. On the other hand, if an optimization problem takes fixed user association as input, the delegation process should consider the association of users. For example, if a user is associated with one BS, its delegation should be at or directly connected to this BS to facilitate information aggregation for this user; if multiple BSs serve the same user as in massive MIMO networks, the one with maximum signal strength can be selected.

F. Primal Feasibility Transformation

In the above, the obtained primal solution in each iteration may not be feasible. To better utilize these intermediate solutions, they need to be transformed to solutions that obey the primal constraints. We call this process *primal feasibility transformation* (PFT). The obvious merit of PFT is to gradually improve network performance during the optimization process (which can be pretty long due to the intrinsic complexity of wireless optimization), without waiting for convergence.

Since the primal constraint set is convex, many convex optimization methods can be used for PFT. However, we seek for a distributed method that does not add much overhead to the above iterative method. We propose the following 2-step PFT method: first, we use the averaging scheme in [**?**] to obtain an approximate feasible solution, whose total feasibility violation is bounded as shown in [**?**]; then, we use *linear scaling* to further transform it to a feasible solution.

Specifically, consider the PFT conducted after iteration i. We first obtain the average of each primal variable over the first i iterations. Denote the average using vector $\hat{\boldsymbol{z}}^{(i)}$, we have

$$\hat{\boldsymbol{z}}^{(i)} = \frac{1}{i}\sum\nolimits_{j=0}^{i} \boldsymbol{z}^{(j)}, \tag{33}$$

where $\boldsymbol{z}^{(0)}$ is the initial point.

Since $\hat{\boldsymbol{z}}^{(i)}$ still may not be feasible, we further scale the primal variables to enforce the feasibility constraints. Observe that all the primal constraints are linear except Constraint (**??**), which is a summation of linear functions projected onto the non-negative real number set. Also observe that variables $\hat{\boldsymbol{x}}^{(i)}$ and $\hat{\boldsymbol{\alpha}}^{(i)}$'s bounds are already satisfied in Π due to the convexity of Π, and are only used to bound the radio resource allocations $\hat{\boldsymbol{a}}^{+,(i)}$ and $\hat{\boldsymbol{a}}^{-,(i)}$. On the other hand, the rates $\hat{\boldsymbol{r}}^{(i)}$ are also determined by allocation variables $\hat{\boldsymbol{a}}^{+,(i)}$ and $\hat{\boldsymbol{a}}^{-,(i)}$. Therefore, we only need to scale variables $\hat{\boldsymbol{a}}^{+,(i)}$ and $\hat{\boldsymbol{a}}^{-,(i)}$ based on violation of Constraints (**??**), (**??**), (**??**), (**??**), and (**??**). The scaling is per-BS based, and is shown as follows:

1) For each BS B and user $U \in \mathcal{U}_B$, if $\hat{a}^{+,(i)}_{B,U} + \hat{a}^{-,(i)}_{B,U} > A \cdot \hat{x}^{(i)}_{B,U}$, then we multiply both $\hat{a}^{+,(i)}_{B,U}$ and $\hat{a}^{-,(i)}_{B,U}$ by $A \cdot \hat{x}^{(i)}_{B,U}/(\hat{a}^{+,(i)}_{B,U} + \hat{a}^{-,(i)}_{B,U})$.
2) For each SBS B, if $\sum_{U\in\mathcal{U}_B} \hat{a}^{-,(i)}_{B,U} > A \cdot \hat{\alpha}^{(i)}_{M(B)}$, we multiply $\hat{a}^{-,(i)}_{B,U}$ for each $U \in \mathcal{U}_B$ by $(A \cdot \hat{\alpha}^{(i)}_{M(B)})/(\sum_{U\in\mathcal{U}_B} \hat{a}^{-,(i)}_{B,U})$; for MBS B, we set $\hat{a}^{-,(i)}_{B,U} = 0$ for $U \in \mathcal{U}_B$.
3) For each BS B, if $\sum_{U\in\mathcal{U}_B} \hat{a}^{+,(i)}_{B,U} > A \cdot (1 - \hat{\alpha}^{(i)}_{M(B)})$, we multiply $\hat{a}^{+}_{B,U}$ for each $U \in \mathcal{U}_B$ by $(A \cdot (1 - \hat{\alpha}^{(i)}_{M(B)}))/(\sum_{U\in\mathcal{U}_B} \hat{a}^{+,(i)}_{B,U})$.
4) For each BS B, if $\sum_{U\in\mathcal{U}_B} (\eta^{+}_{B,U}\hat{a}^{+,(i)}_{B,U} + \eta^{-}_{B,U}\hat{a}^{-,(i)}_{B,U}) > \beta_B$, we multiply $\hat{a}^{+}_{B,U}$ and $a^{-}_{B,U}$ for each $U \in \mathcal{U}_B$ by $\beta_B/(\sum_{U\in\mathcal{U}_B} (\eta^{+}_{B,U}\hat{a}^{+,(i)}_{B,U} + \eta^{-}_{B,U}\hat{a}^{-,(i)}_{B,U}))$.
5) In the global view of the network, let $\mathcal{P}^{vio} = \sum_{B\in\mathcal{B}} \left[\rho^f_B + \rho^t_B \cdot \sum_{U\in\mathcal{U}_B} (\hat{a}^{+,(i)}_{B,U} + \hat{a}^{-,(i)}_{B,U}) - \rho^n_B\right]^+ - \mathcal{P}$. If $\mathcal{P}^{vio} > 0$, we need to scale resources. Note that both the fixed power consumption and the part of radio resources covered by renewable power cannot be scaled, thus we compute the scalable power consumption per BS as $\mathcal{P}^{scale}_B = \rho^t_B \sum_{U\in\mathcal{U}_B} (\hat{a}^{+,(i)}_{B,U} + \hat{a}^{-,(i)}_{B,U}) - [\rho^n_p - \rho^f_p]^+$. We multiply each BS B's total resources (thus $\hat{a}^{+}_{B,U}$ and $\hat{a}^{-}_{B,U}$ for $\forall U \in \mathcal{U}_B$) by $\left(1 - \frac{\mathcal{P}^{scale}_B}{(\rho^t_B \sum_{U\in\mathcal{U}_B} (\hat{a}^{+,(i)}_{B,U} + \hat{a}^{-,(i)}_{B,U}))} \cdot \frac{\mathcal{P}^{vio}}{\sum_{B\in\mathcal{B}} \mathcal{P}^{scale}_B}\right)$, which enforces the total power constraint.

After the above, the obtained resource allocations $\boldsymbol{a}^+$ and $\boldsymbol{a}^-$ satisfy Constraints (**??**), (**??**), (**??**), (**??**), and (**??**). The rate of each user $U \in \mathcal{U}$ is then re-computed as

$$\hat{r}^{(i)}_U = \sum\nolimits_{B\in\mathcal{B}_U} (\eta^{+}_{B,U}\hat{a}^{+,(i)}_{B,U} + \eta^{-}_{B,U}\hat{a}^{-,(i)}_{B,U}). \tag{34}$$

The obtained solution $\hat{\boldsymbol{z}}^{(i)}$ is now primal feasible.

Storage: Additional $\sum_{B\in\mathcal{S}_M\cup\{M\}} |\mathcal{U}_B|$ variables at each LC, as shown below.

Communications: The averaging process does not involve communications. The scaling process mostly uses local information stored at each BS delegate, except $\hat{\boldsymbol{x}}^{(i)}_{B,U}$ needed from each user delegate in step 1, and the ratio $\mathcal{P}^{vio}/\sum_{B\in\mathcal{B}} \mathcal{P}^{scale}_B$ from the GC in step 5. Since the variables $\boldsymbol{x}^{(i)}_{B,U}$ are exchanged per-iteration for dual update, each BS delegate can keep a copy of their average values, adding $\sum_{B\in\mathcal{S}_M\cup\{M\}} |\mathcal{U}_B|$ to the storage of each LC while eliminating the communications of another round of x variables. Hence per iteration, each LC only aggregates total scalable power for all delegated BSs, and sends it to the GC; the GC then broadcasts the ratio $\mathcal{P}^{vio}/\sum_{B\in\mathcal{B}} \mathcal{P}^{scale}_B$ to all LCs. The communication overhead at each LC is 2 messages (1 input and 1 output), and that at the GC is $2|\mathcal{M}|$ messages.

Since the dual subgradient method is not a descent method, the current feasible solution may be worse than the best feasible solution ever found. Hence the GC records the best feasible solution, and instructs the LCs to execute it until improved in the future. This adds a copy of local primal variables of the best solution at each controller, and another 2 messages between each LC and the GC, one for reporting aggregated objective value, another for informing the better solution between current and previous best.

Remark 4.3: The above PFT does not affect the convergence of the optimization. Moreover, as proved in [**?**], if

the optimization converges to the optimal solution, so does the averaging sequence. Since linear scaling does not affect convergence, the primal solution after PFT also converges to the optimal. The pros of using PFT is to improve average network objective by utilizing intermediate solutions. Yet the cost is the additional overhead, for example, the additional running time per iteration, and the extra storage/communication overhead on each controller. In this sense, a PFT method should have low complexity in order not to add non-negligible overhead to the optimization. For example, the above PFT method runs in linear time, uses linear space, and adds minimal communication overhead per controller.

Remark 4.4: While we have studied a specific optimization problem in RAN as an illustrative example, the same architecture and technique can be applied to dealing with various optimization problems and methods in the RAN. Specifically, implementing distributed optimization in HSDRAN essentially involves three steps: 1) finding a distributed/decomposition-based algorithm, 2) defining computation delegation to local controllers, and 3) finding a PFT method and also define its delegation scheme. Many methods can be used for step 1, including dual subgradient method, ADMM [**?**], etc. In step 2, an efficient computation delegation scheme should follow the principle so as to minimize the communication overhead between controllers, and can be analytically derived from the optimization method found in step 2. The PFT step is optional, and is only useful when the method in step 1 does not guarantee primal feasibility during iterations.

Remark 4.5: In the above optimization, we assume the RAN is operated by a single service provider. In a multi-operator network [**?**], each operator may have a different set of network objective and constraints, and network resources are dynamically shared among different operators. Our architecture and optimization framework can be used orthogonally with network slicing techniques such as RadioVisor [**?**]. In this case, each operator initiates its own HSDRAN control plane, with both the GC and the LCs in its own allocated network slice. The global network slice manager, which has a higher hierarchy than the operators' GCs, will make dynamic radio resource allocation for each operator. This allocation will then provide the available radio resources to each operator as input to our HSDRAN optimization framework. Modeling and coordinating multi-operator competition is a different problem than the one we tackle, and is among our future research directions.

V. Performance Evaluation

A. Experiment Settings

We implemented our proposed optimization under both centralized SDRAN and HSDRAN using Matlab, and compared their overhead. General default experiment parameters are listed in Table **??**. Parameters marked as "varying" are varied in the experiments. Fig. **??** shows an example of the topology. 3 MBSs are deployed by default, as shown by MBS 1–3 in the figure, each with 4 randomly deployed SBSs. When increasing MBSs, each MBS is added in order of 1–7; MBSs excessing 7 are added in the same manner, *i.e.*, in clockwise sense around inner MBSs. We considered two types of users: *normal users* that are uniformly distributed in the simulated area, and *clustered users* that are within 40m of an arbitrary SBS.

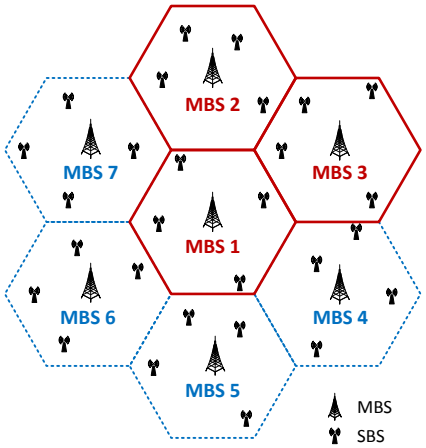


Fig. 3: Experiment topology where MBSs 1–3 are by default, and MBSs 4–7 and beyond are added in some experiments. SBSs are randomly deployed within each MBS.

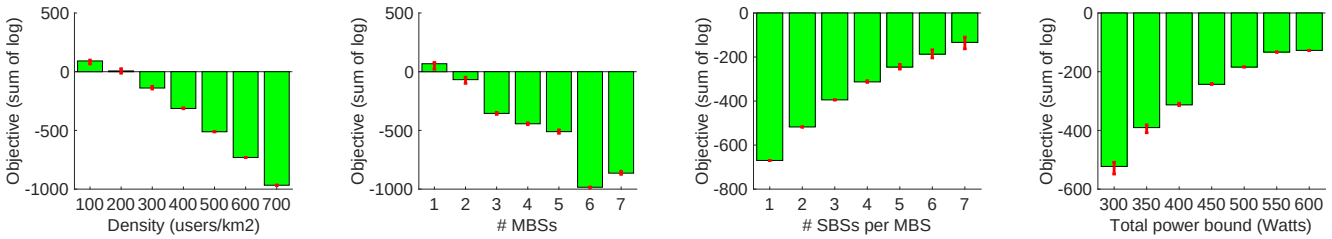
TABLE IV: General Experiment Parameters

# MBSs	3 (varying)
# SBSs per MBS	4 (varying)
Inter-site distance [**?**]	500 m
User density [**?**]	400 /km^2 (varying)
Clustered user ratio [**?**]	$2/3$
Total radio bandwidth ($\mathcal{A}$) [**?**]	10 Mbps
Network-wide power bound ($\mathcal{P}$)	400 Watts (varying)
Noise [**?**]	-174 dBm/Hz
Minimum SINR (Υ)	-10 dB (varying)

Default wireless, backhaul and power parameters are shown in Table **??**. In the experiments, we varied different parameters. Each experiment ran the iterative algorithm for 10000 iterations, with an initial point that distributes radio resources evenly among users. Step size of 0.0007 is empirically chosen. In each iteration, we utilized the PFT in Section **??** to obtain a feasible solution, executed it if better than the current, and recorded the average objective throughout the optimization. We ran each experiment for 20 times under the same setting and took the average. Experiments were run on a Macbook Air with Intel Core i7 1.7GHz CPU and 8GB memory.

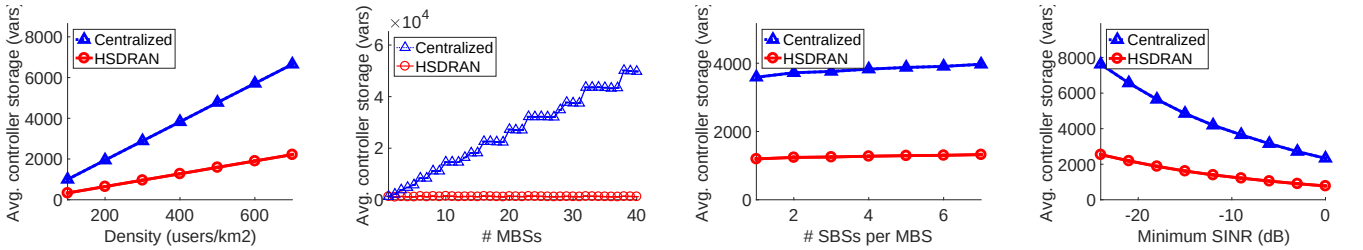
TABLE V: Wireless, Backhaul and Power Parameters for BSs

	MBS	**SBS**
Path loss (d in km) [**?**]	$37.6 \log_{10}(d) + 128.1$	$36.7 \log_{10}(d) + 140.7$
Transmit power [**?**]	46 dBm	30 dBm
Backhaul [**?**]	2000 Mbps	200 Mbps
Fixed power cons. [**?**]	780 Watts	13.6 Watts
Trans. power slope [**?**]	18.711 Watts/MHz	0.4 Watts/MHz
Renew. power source [**?**]	696 Watts	15.66 Watts



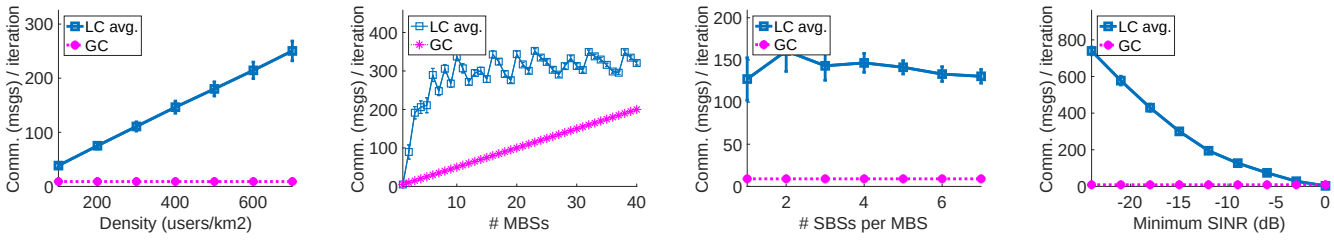
(a) Network objective vs. user density (b) Network objective vs. MBSs (c) Network objective vs. SBSs / MBS (d) Network objective vs. power bnd

Fig. 4: Network objectives with varying user densities, MBSs, SBSs per MBS and total power bounds.



(a) Controller storage vs. user density (b) Controller storage vs. MBSs (c) Controller storage vs. SBSs / MBS (d) Controller storage vs. min. SINR

Fig. 5: Average controller storage overhead with varying user densities, MBSs, SBSs per MBS and minimum SINR.



(a) Comm. vs. user density (b) Comm. vs. MBSs (c) Comm. vs. SBSs per MBS (d) Comm. vs. min. SINR

Fig. 6: Average controller communication overhead with varying user densities, MBSs, SBSs per MBS and minimum SINR.

B. Experiment Results

Fig. **??** shows the network objectives with varying user densities, MBSs, SBSs per MBS and total power bounds. Bars show the average objective values over all iterations. Error bars ("I"-like lines) show the initial objective values and the final objective values as lower and upper bounds respectively. The change of objective values follows intuitive analysis, where the network objective decreases when users or MBSs increase (the latter due to decreased per-MBS power), and increases when number of SBSs or power bound increases (both due to more available resources). In many cases, the average network objective is noticeably higher than the initial network objective, for example, when user density, number of MBSs or power bound is low, or when number of SBSs is high. We also observe that the final network objective is noticeably higher than the initial objective in these cases. This shows that our proposed PFT indeed increases average network objective.

Fig. **??** shows the average per-controller storage overhead (number of variables) with varying user densities, MBSs, SBSs per MBS and minimum SINR requirement. We compare our HSDRAN implementation to a fully centralized implementation where all computations are conducted at the GC. In the network, the number of users dominates the number of BSs, hence the storage overhead grows linearly with user density. On the other hand, while more MBSs greatly increases storage in the centralized implementation, it almost has no impact on HSDRAN, which well illustrates how HSDRAN achieves load balancing and scalability. While the number of SBSs indeed has some impact on storage, the impact is minimal, given that most users receive strong-enough signal from at most one SBS. Finally, storage decreases as minimum SINR increases, due to

that fewer BSs are in the candidate set of each user. In all cases, the per-controller storage overhead is much lower in HSDRAN than in the centralized implementation. Note that we also plotted the 95% confidence intervals for the 20 times of tests. However, the scale of the confidence intervals are too small compared to the mean value, hence cannot be well visible from the figures. In average, the scale of the 95% confidence intervals are within 0.5% of the mean values.

Fig. **??** shows the per-controller communication overhead with varying user densities, MBSs, SBSs per MBS and minimum SINR requirement. No inter-controller communication is needed in the centralized implementation. We show both the average per-LC communications, and the communications at the GC. In all cases, the GC incurs very little communication overhead. LC communications increase with user density due to more exchanged information, while GC communications remain the same. On the other hand, more MBSs does not always increase LCs' communications; the LC at each MBS only communicates with LCs at nearby MBSs, hence the communications reach a balance point after a certain amount of increase. This validates that HSDRAN utilizes locality to reduce communication overhead, leading to great scalability and load balancing advantages. GC communications increase linearly with the number of MBSs (thus LCs), which matches our analysis. The number of SBSs basically does not affect communications, as each user typically receives strong-enough signal from at most one SBS. With increasing SINR threshold, LC communications go down drastically due to less chance that a user receives signals from two MBSs (or associated SBSs). This shows how locality is crucial in reducing communication overhead. We also plotted the 95% confidence intervals for LC average communication overhead; GC communication is only dependent on the number of LCs, hence remains the same for all the 20 runs. In average, the scale of the 95% confidence intervals are within 3% of the mean values.

To summarize, HSDRAN is able to balance load and reduce overhead (e.g. storage) at each controller, avoiding single-point bottleneck at the central controller. It utilizes network locality to limit communication overhead incurred at each controller. With PFT, it can also improve average network performance compared to using the initial solution until convergence.

Remark 5.1: While HSDRAN features its distributed control plane, fully centralized (global) control applications can still be readily deployed at the GC in HSDRAN. In fact, by offloading locality-aware applications such as distributed optimization, the GC can dedicate more computation power to global applications. HSDRAN can thus be viewed as a performance-enhancing extension to centralized SDRAN.

VI. Discussions

Online optimization: Network dynamics happen frequently in RANs due to user mobility, channel fluctuation, etc. While our proposed framework is for static optimization, it can be applied to online optimization with modifications. To respond to network dynamics, the LCs need to aggregate real-time network statistics from the data plane. When dynamics happen within an LC's control domain, they will be accounted for in future iterations of the optimization. Using our delegation scheme, the added storage and communication overhead can be close to none, because only results of each iteration need to be exchanged. In practice, the optimization may not converge due to frequent dynamics. Applying PFT, the network can always benefit from the optimization instead of waiting for convergence. However, the current averaging-based PFT method is not suitable for frequent network dynamics. Advanced online optimization and PFT methods are within our future work.

Multi-layer control plane: HSDRAN has a two-layer control plane. In the future, more complex communication models need more layers for further scalability and load balancing. E.g., powerful SBSs can host another layer of LCs for D2D communications [**?**], vehicular communications, or wireless sensor networks [**?**]. Advanced distributed optimization and offloading techniques are in need for this architecture.

Logically centralized control plane: In traditional SDN, scalability is achieved by the so-called *logically centralized* control plane, where multiple copies of the global controller are deployed for load balancing. However, each controller still needs to keep a global view, hence it cannot reduce storage and information aggregation overhead, and also incurs communications between controllers. On the contrary, HSDRAN leverages locality in RANs, letting each LC maintain only local information, while the GC only receives and stores aggregated information from LCs. This effectively reduces storage and communication overhead at any controller.

VII. Conclusions

In this paper, we proposed an HSDRAN architecture that achieves self-optimization, scalability, load balancing, and responsiveness at the same time. The architecture deploys local controllers in additional to the global controller to offload control tasks to the network edge. Besides local decision making and execution of global decisions, the local controllers also participate in globally coordinated network optimization. We then presented a framework to implement optimization in HSDRAN. We used a paradigmatic user association and downlink resource allocation problem to illustrate the benefits of our architecture and framework. The problem was solved using a distributed dual subgradient method. Via extensive simulations, we showed how our architecture and framework can improve average network objective with balanced storage usage at each controller and limited inter-controller communications.

References

[1] 3GPP, "Further Advancements for E-UTRA, Physical Layer Aspects (Release 9)," Tech. Rep., 2010.

[2] H. Ali-Ahmad, C. Cicconetti, A. de la Oliva, V. Mancuso, M. R. Sama, P. Seite, and S. Shanmugalingam, "An SDN-Based Network Architecture for Extremely Dense Wireless Networks," in *IEEE SDN4FNS*, 2013.

[3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, jun 2014.

[4] M. Arslan, K. Sundaresan, and S. Rangarajan, "Software-defined Networking in Cellular Radio Access Networks: Potential and Challenges," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 150–156, 2015.

[5] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A Programmable Wireless Dataplane," in *ACM HotSDN*, 2012.

[6] D. Bethanabhotla, O. Y. Bursalioglu, H. C. Papadopoulos, and G. Caire, "Optimal User-Cell Association for Massive MIMO Wireless Networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 3, pp. 1835–1850, mar 2016.

[7] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient Methods," 2003. URL: https://web.stanford.edu/class/ee392o/subgrad_method.pdf

[8] Z. Chang, Y. Gu, Z. Han, X. Chen, and T. Ristaniemi, "Context-aware Data Caching for 5G Heterogeneous Small Cells Networks," in *IEEE ICC*, 2016.

[9] H. Chen and W. Lou, "On Protecting End-To-End Location Privacy Against Local Eavesdropper in Wireless Sensor Networks," *Pervasive Mob. Comput.*, vol. 16, no. PA, pp. 36–50, jan 2015.

[10] X. Chen, Z. Han, H. Zhang, M. Bennis, and T. Chen, "Foresighted Resource Scheduling in Software-Defined Radio Access Networks," in *IEEE GlobalSIP*, 2015, pp. 128–132.

[11] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-Efficiency Oriented Traffic Offloading in Wireless Networks: A Brief Survey and a Learning Approach for Heterogeneous Cellular Networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 627–640, apr 2015.

[12] M. Cierny, H. Wang, R. Wichman, Z. Ding, and C. Wijting, "On Number of Almost Blank Subframes in Heterogeneous Cellular Networks," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 10, pp. 5061–5073, oct 2013.

[13] R. Cwalinski and H. Koenig, "RADIator - An Approach for Controllable Wireless Networks," in *IEEE NetSoft*, 2016, pp. 260–268.

[14] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for Enhanced Inter-Cell Interference Coordination (eICIC) in LTE HetNets," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 137–150, feb 2014.

[15] X. Duan and X. Wang, "Authentication Handover and Privacy Protection in 5G Hetnets Using Software-Defined Networking," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 28–35, apr 2015.

[16] A. A. Gebremariam, L. Goratti, R. Riggio, D. Siracusa, T. Rasheed, and F. Granelli, "A Framework for Interference Control in Software-Defined Mobile Radio Networks," in *IEEE CCNC*, 2015, pp. 892–897.

[17] F. Granelli, A. A. Gebremariam, M. Usman, F. Cugini, V. Stamati, M. Alitska, and P. Chatzimisios, "Software Defined and Virtualized Wireless Access in Future Wireless Networks: Scenarios and Standards," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 26–34, jun 2015.

[18] A. Gudipati, L. E. Li, and S. Katti, "RadioVisor: A Slicing Plane for Radio Access Networks," in *ACM HotSDN*, 2014, pp. 237–238.

[19] A. Gudipati, D. Perry, L. E. Li, S. Katti, and B. Labs, "SoftRAN: Software Defined Radio Access Network," in *ACM HotSDN*, 2013, pp. 25–30.

[20] Q. Han, B. Yang, G. Miao, C. Chen, X. Wang, and X. Guan, "Backhaul-Aware User Association and Resource Allocation for Energy-Constrained HetNets," *IEEE Trans. Veh. Technol.*, vol. PP, no. c, 2016.

[21] Z. Han, D. Niyato, W. Saad, T. Baar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, 2012.

[22] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in *ACM CoNEXT*, 2013, pp. 163–174.

[23] F. K. Jondral, "Software-defined Radio: Basics and Evolution to Cognitive Radio," *EURASIP J. Wirel. Commun. Netw.*, vol. 2005, no. 3, 2005.

[24] J. Kerttula, N. Malm, K. Ruttik, R. Jäntti, and O. Tirkkonen, "Implementing TD-LTE as Software Defined Radio in General Purpose Processor," in *ACM SRIF*, 2014, pp. 61–68.

[25] W.-C. Liao, M. Hong, H. Farmanbar, X. Li, Z.-Q. Luo, and H. Zhang, "Min Flow Rate Maximization for Software Defined Radio Access Networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1282–1294, jun 2014.

[26] D. Liu, Y. Chen, K. K. Chai, T. Zhang, and K. Han, "Joint User Association and Green Energy Allocation in HetNets with Hybrid Energy Sources," in *IEEE WCNC*, 2015, pp. 1542–1547.

[27] Y. Liu, C. S. Chen, and C. W. Sung, "Joint Optimization on Inter-Cell Interference Management and User Attachment in LTE-A HetNets," in *IEEE WiOpt*, 2015, pp. 62–69.

[28] M. Moradi, W. Wu, L. E. Li, and Z. M. Mao, "SoftMoW: Recursive and Reconfigurable Cellular WAN Architecture," in *ACM CoNEXT*, 2014, pp. 377–390.

[29] A. Nedić and A. Ozdaglar, "Approximate Primal Solutions and Rate Analysis for Dual Subgradient Methods," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1757–1780, jan 2009.

[30] Panasonic, "HIT Power 220A." URL: http://www.panasonic.com/business/pesna/includes/pdf/eco-construction-solution/HIT_Power_220A_Datasheet.pdf

[31] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed Multi-Domain SDN Controllers," in *IEEE NOMS*, 2014.

[32] R. Riggio, K. Gomez, L. Goratti, R. Fedrizzi, and T. Rasheed, "V-Cell: Going Beyond the Cell Abstraction in 5G Mobile Networks," in *IEEE NOMS*, 2014.

[33] M. A. S. Santos, B. A. A. Nunes, K. Obraczka, T. Turletti, B. T. de Oliveira, and C. B. Margi, "Decentralizing SDN's Control Plane," in *IEEE LCN*, 2014, pp. 402–405.

[34] S. Schmid and J. Suomela, "Exploiting Locality in Distributed SDN Control," in *ACM HotSDN*, 2013.

[35] P. Spapis, K. Chatzikokolakis, N. Alonistioti, and A. Kaloxylos, "Using SDN as a Key Enabler for Co-Primary Spectrum Sharing," in *IEEE IISA*, 2014, pp. 366–371.

[36] J. Wang, H. Jiang, Z. Pan, N. Liu, X. You, and T. Deng, "Joint User Association and ABS Proportion Optimization for Load Balancing in HetNet," in *IEEE WCSP*, 2015.

[37] W. Wu, L. E. Li, A. Panda, and S. Shenker, "PRAN: Programmable Radio Access Networks," in *ACM HotNets*, 2014.

[38] S. H. Yeganeh and Y. Ganjali, "Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications," in *ACM HotSDN*, 2012.

[39] R. Yu, S. Qin, M. Bennis, X. Chen, G. Feng, Z. Han, and G. Xue, "Enhancing Software-Defined RAN with Collaborative Caching and Scalable Video Coding," in *IEEE ICC*, 2016.

[40] Z. Zaidi, V. Friderikos, and M. A. Imran, "Future RAN Architecture: SD-RAN Through a General-Purpose Processing Platform," *IEEE Veh. Technol. Mag.*, vol. 10, no. 1, pp. 52–60, mar 2015.

[41] T. Zhao, L. Wang, X. Zheng, S. Zhou, and Z. Niu, "HyCell: Enabling GREEN Base Station Operations in Software-Defined Radio Access Networks," in *IEEE ICCW*, 2015, pp. 2868–2873.

[42] H. Zhou, Y. Ji, X. Wang, and S. Yamada, "Joint Spectrum Sharing and ABS Adaptation for Network Virtualization in Heterogeneous Cellular Networks," in *IEEE GLOBECOM*, 2015.

[43] H. Zhou, Y. Ji, X. Wang, and B. Zhao, "ADMM Based Algorithm for eICIC Configuration in Heterogeneous Cellular Networks," in *IEEE INFOCOM*, 2015, pp. 343–351.

[44] M. Zhou, H. Zhang, S. Zhang, L. Song, Y. Li, and Z. Han, "Design and Implementation of Device-To-Device Software-Defined Networks," in *IEEE ICC*, 2016.

Ruozhou Yu (Student Member 2013) received his B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2013. Currently he is a Ph.D student in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University. His research interests include network virtualization, software-defined networking, and cloud and data center networks.

Guoliang Xue (Member 1996, Senior Member 1999, Fellow, 2011) is a professor of Computer Science and Engineering at Arizona State University. He received the PhD degree in Computer Science from the University of Minnesota in 1991. His research interests span the areas of Quality of Service provisioning, network security and privacy, crowdsourcing and network economics, RFID systems and Internet of Things, smart city and smart grids. He has published over 280 papers in these areas, many of which in top conferences such as INFOCOM, MOBICOM, NDSS and top journals such as IEEE/ACM Transactions on Networking, IEEE JSAC, IEEE TMC. He was a keynote speaker at IEEE LCN'2011 and ICNC'2014. He was a TPC Co-Chair of IEEE INFOCOM'2010 and a General Co-Chair of IEEE CNS'2014. He has served on the TPC of many conferences, including ACM CCS, ACM MOBIHOC, IEEE ICNP, and IEEE INFOCOM. He served on the editorial board of IEEE/ACM Transactions on Networking. He serves as the Area Editor of IEEE Transactions on Wireless Communications, overseeing 13 editors in the Wireless Networking area. He is an IEEE Fellow, and the VP-Conferences of the IEEE Communications Society.

Mehdi Bennis (Senior Member, IEEE) received his M.Sc. degree in Electrical Engineering jointly from the EPFL, Switzerland and the Eurecom Institute, France in 2002. From 2002 to 2004, he worked as a research engineer at IMRA-EUROPE investigating adaptive equalization algorithms for mobile digital TV. In 2004, he joined the Centre for Wireless Communications (CWC) at the University of Oulu, Finland as a research scientist. In 2008, he was a visiting researcher at the Alcatel-Lucent chair on flexible radio, SUPELEC. He obtained his Ph.D. in December 2009 on spectrum sharing for future mobile cellular systems. Currently Dr. Bennis is an Adjunct Professor at the University of Oulu and Academy of Finland research fellow. His main research interests are in radio resource management, heterogeneous networks, game theory and machine learning in 5G networks and beyond. He has co-authored one book and published more than 100 research papers in international conferences, journals and book chapters. He was the recipient of the prestigious 2015 Fred W. Ellersick Prize from the IEEE Communications Society and the 2016 Best Tutorial Prize from the IEEE Communications Society. Dr. Bennis serves as an editor for the IEEE Transactions on Wireless Communication.

Xianfu Chen (Member, IEEE) received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2012. He is currently a Senior Scientist at VTT Technical Research Centre of Finland Ltd., Oulu, Finland. His research interests cover various aspects of wireless communications and networking, with emphasis on software-defined radio access networks, green communications, centralized and decentralized resource allocation, and the application of artificial intelligence to wireless communications.

Zhu Han (Student Member 2001, Member 2004, Senior Member 2009, Fellow 2014) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Currently, Dr. Han is an IEEE Communications Society Distinguished Lecturer.