AppRAN: Application-Oriented Radio Access Network Sharing in Mobile Networks

Jun He and Wei Song
Faculty of Computer Science
University of New Brunswick, Fredericton, Canada
Emails: {jhe2, wsong}@unb.ca

***Abstract*—As a promising way to increase network capacity and reduce expenses, radio access network (RAN) sharing among mobile (virtual) network operators, has attracted extensive recent attention from both industry and academia. Meanwhile, mobile systems are undergoing fast evolution to virtualized infrastructure so as to tackle the ever-growing mobile traffic and the unremitting demand for high data rates. However, existing RAN sharing models intend to expose resource details, *e.g.,* infrastructure and spectrum, to participating network operators of the RAN for resource-sharing purposes, which violates the principles of network abstraction and makes network management even more complicated. This paper presents AppRAN, an application-oriented framework for RAN sharing in mobile networks, which decouples network operators from radio resource by providing application-level services with Quality of Service (QoS) guarantee. AppRAN defines a serial of abstract applications with distinct QoS requirements and periodically computes application-level resource allocation for each radio element at a central controller w.r.t. traffic demands and average channel condition. The radio elements are allowed to independently determine flow-level resource allocation within each application afterwards. We formulate the application-level resource allocation as an optimization problem and develop a fast algorithm to solve it with a provably approximate guarantee. The efficacy of AppRAN is validated through theoretical analysis and computer simulations. We show that AppRAN is in line with the design of software-defined RAN.**

***Index Terms*—Radio access network, RAN sharing, software-defined RAN, resource virtualization, network abstraction**

I. Introduction and Related Work

Nowadays, mobile network operators (MNOs) are increasingly facing up to the realities of unremitting demands for high date rate and continuously declining unit-data revenue. Thousands of applications settle in mobile networks and provide services to data-hungry devices owned by customers who increasingly consider ubiquitous Internet access as a human right regardless of the overburdening of networks and the high costs to upgrade systems. To expand system capacity and reduce both capital expenses (CAPEX) and operational expenses (OPEX), MNOs tend to share their infrastructure with each other and other service providers (SPs) in various forms, among which radio access network (RAN) sharing has the most impact [1,2]. In addition, service-oriented entities, such as mobile virtual network operators (MVNOs), content providers (CPs), are emerging RAN sharing participants, who do not own infrastructure but rent resource or capacity from MNOs. However, RAN sharing is a challenging task.

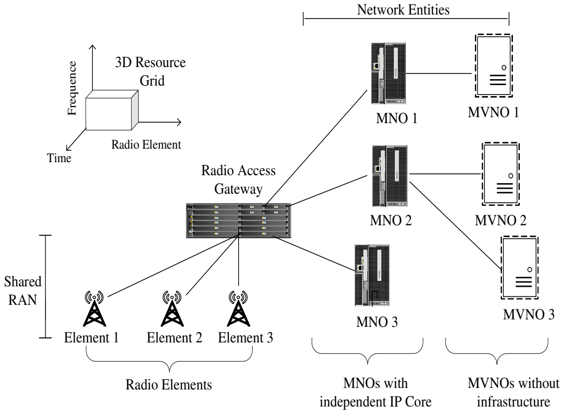


Figure 1. RAN Sharing Network Model.

The oldest form of RAN sharing in MNOs is to allow mobile access of foreign subscribers roaming from other networks [3]. The 3rd Generation Partnership Project (3GPP) network sharing architecture for the Long-Term Evolution (LTE) enables different core network operators to connect to a shared RAN in either a gateway core network (GWCN) configuration or a multi-operator core network (MOCN) configuration, with shared or independent mobility management entities (MMEs), respectively [4]. Entities like MVNOs are actually CPs or resellers, who share RAN in a rental manner based on their service level agreements (SLAs) with MNOs. Along with the proliferation of smart devices, these entities enrich mobile networks with innovative applications, differentiated services, and promote subscribers' engagements [5] and are increasing the share of mobile networks rapidly [6]. Figure 1 shows the network model of sharing RAN we study in this paper. MNOs share the RAN through a radio access gateway, *e.g.,* the serving gateway in LTE or the access service network (ASN) gateway in WiMAX, and provide RAN access to MVNOs via their IP cores. Radio elements refer to base-stations, *e.g.,* eNodeB, pico/micro cells, that are managed by a centralized controller at the RAN gateway and provide radio access to subscribers.

As mobile networks are merging into the cloud, RANs are undergoing fast evolution to network virtualization [7]. With the promotion of scalability and manageability, virtualized RANs have developed maturing methods for resource slicing

This research was supported in part by Mariner Partners Inc., Atlantic Canada Opportunities Agency, and Natural Sciences and Engineering Research Council (NSERC) of Canada.

and frame scheduling, which eases the resource management at the controller [8]–[10]. As depicted in Figure 1, the radio resource over the RAN is abstracted in a configurable *3-dimensional resource grid* of radio element index, frequency and time [8,9]. The central controller then has a view of one virtual "big" base-station upon which radio resource is slicable and allocatable via the northbound application programming interfaces (APIs) that the virtualized RAN provides to the controller.

State-of-the-art resource schedulers, *e.g.,* [10,11], divide resource among entities sharing the RAN based on SLAs with entity isolation in a resource-reservation manner. That is, SLAs specify the resource shares of each entity either on a per-base-station basis (*e.g.,* [10]) or on a RAN basis (*e.g.,* [11]). For instance, in a network with 2 entities, entity 1 reserves 30% of the resource (overall or per-base-station) while entity 2 takes 70%. These share ratios could also be a range, *e.g.,* minimal 20% and maximal 35%, to enable adaptive resource scheduling according to data traffic [11]. The allocation decisions made at the controller are then applied by lower-layer frame schedulers. However, due to the following concerns, we argue that these entity-oriented designs are against the principles of network virtualization and will soon become infeasible in the expanding mobile networks.

- These methods expose extensive details of RAN to the entities, *e.g.,* the number, distribution and capacity of radio elements. Even though the resource share is given in percentages, for pricing and budget purposes, one entity needs to know the coverage of the RAN and the bandwidth it provides. Therefore, these entity-oriented approaches will make the network management and RAN upgrade even more complicated as the RAN or the number of entities grows.
- To fulfill SLAs, the controller requires the ownership information of each data flow, which can only be retrieved by conducting deep packet inspections (DPIs). Apparently, the overhead of DPI per flow would be intolerable.
- It is hard to manage QoS in these models. Each entity provides a set of services (regarded as applications hereafter) with differentiated QoS requirements. Since the resource is allocated at the entity level, each entity independently allocates its resource to applications for QoS management, *e.g.,* in the model of bearers [12,13], which not only makes QoS support at the RAN gateway more complicated, but also produces an aggregate resource utility arbitrarily suboptimal from the application view.
- These existing works determine detailed resource allocation (*e.g.,* per flow) at the central controller, which is incompatible to new developments in RAN with heterogeneous radio elements due to the overwhelming overhead of reporting wireless channel details. For example, only symmetric base-stations are considered in [11].

To address these problems, in this paper, we re-define the RAN sharing model and propose AppRAN, an application-oriented resource-sharing framework. AppRAN promises to ease network management as well as promote resource utility by decoupling entities from radio resource allocation. Entities can now focus on their application-level demands through an application-abstraction layer provided by AppRAN, while the framework takes care of lower-layer resource management.

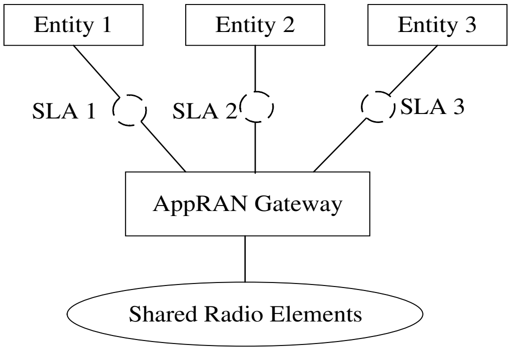


Figure 2. AppRAN Network Model.

In summary, the contributions of this paper are three-fold. Firstly, to our best knowledge, this work is the first attempt towards application-oriented RAN sharing in mobile networks. AppRAN decouples entities from radio resource allocation and provides a better network abstraction. Instead of bringing the resource "cake" to the table and splitting it in the presence of all entities, AppRAN promises the QoS of abstract applications it supports such that entities can map their concrete applications to the abstract ones and determine application-specified bandwidth they need. AppRAN, therefore, keeps the upgrade of RAN facilities and resource allocation transparent to entities and enables better resource virtualization.

Secondly, AppRAN defines a new model of service level agreements. In AppRAN, entities are all regarded as clients of the RAN. The charging policy is related to the service package each entity purchases, *e.g.,* in the form of a serial of (application, bandwidth) tuples. Such application-oriented SLA model makes a step towards merging RANs into the cloud.

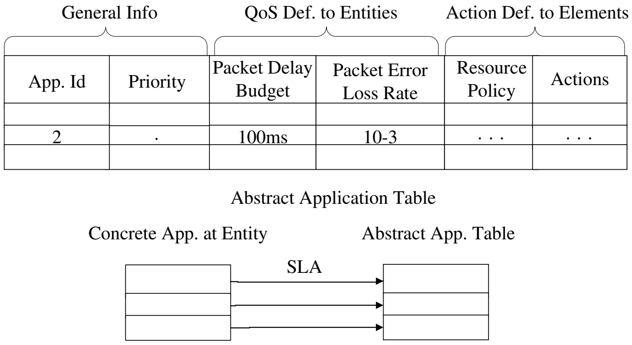
Thirdly, we develop an optimization framework for resource allocation as the kernel of AppRAN and propose a fast algorithm with a provably approximation guarantee. The optimization framework takes average resource-to-bandwidth conversion ratios reported by radio elements as input and computes the optimal resource allocation among applications for each radio element. With negotiable overhead, the central controller then determines the optimal resource allocation on the application level.

The rest of paper is organized as follows. Section II describes the design of AppRAN. The kernel resource allocation algorithm is presented in Section III. Section IV provides the numerical results and Section V concludes the paper.

II. Design of AppRAN

A. The AppRAN Model

AppRAN flattens the RAN sharing network structure and re-models it as Figure 2. AppRAN regards all the network operators sharing the RAN, including MNOs, MVNOs, and CPs, as *entities* driving data flows to the AppRAN gateway, where the flows are differentially treated according to SLA configurations. An MNO with several virtual operators attached to it can be treated as one entity or several entities as it



General Info		QoS Def. to Entities		Action Def. to Elements	
App. Id	Priority	Packet Delay Budget	Packet Error Loss Rate	Resource Policy	Actions
2	·	100ms	10-3	· · ·	· · ·

Figure 3. Definition of Abstract Application and SLA Model.

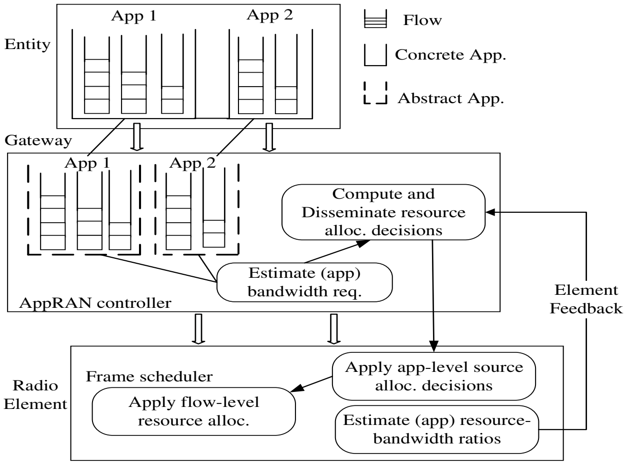
describes, while the details inside are kept transparent to AppRAN, giving the MNO more flexibility.

Instead of exposing resource details to entities, AppRAN defines a serial of *abstract applications* with respect to differentiated QoS levels, which can be readily supported using RAN "bearers" in 3GPP systems [4]. Figure 3 shows an example of the abstract application table. The description of an abstract application consists of identification information (id and priority), QoS guarantees to entities (delay, packet loss rate, *etc.*), action information to radio elements (resource policy, network actions, *etc.*), and (possibly) unit pricing information. An SLA then indicates how to map concrete applications to abstract applications and the bandwidth demand upon each abstract application. AppRAN thus adapts differentiated services [14] supported at respective entities (*e.g.,* bearers in LTE [4]) to the abstract application set. In this way, AppRAN is able to react quickly to new emerging applications, which will become the new norm in future networks [15], by adding entries at the abstract application table. AppRAN also eases network management and resource allocation by abstracting numerous external nonuniform services in a controllable set. As a result, entities are only required to determine the types and the bandwidths of abstract applications they need on a more trackable and readable pricing system produced by the re-modeled SLA.

Based on the bandwidth requirements of abstract applications gathered from SLAs, the AppRAN controller configures a lower bound and an upper bound of resource available to each application to provide isolation among different applications, while the resource within the bounds is adjustable and periodically allocated to each application at a time order of several seconds to promote resource utility according to real-time traffic and wireless channel conditions.

B. The AppRAN Software Architecture

The software architecture of AppRAN is illustrated in Figure 4. Flows of concrete applications in entities are mapped to abstract applications upon entering the RAN gateway according to respective SLAs. Inside the AppRAN gateway, per time period τ (in the order of seconds), the controller estimates the bandwidth requirement of each application, (possibly) through history analysis. On the same periodic scale, each radio element estimates the average resource-bandwidth ratio for each application, *i.e.,* average resource per unit data rate, capturing the average channel condition of corresponding application, and sends these ratios to the controller as element feedback. The rationale behind this is that the resource to support unit data rate is jointly determined by QoS requirements (indicated by types of applications) and wireless channel conditions.



Entity
App 1
App 2
Flow
Concrete App.
Abstract App.
Gateway
App 1
App 2
Compute and Disseminate resource alloc. decisions
Estimate (app) bandwidth req.
AppRAN controller
Element Feedback
Radio Element
Frame scheduler
Apply app-level source alloc. decisions
Apply flow-level resource alloc.
Estimate (app) resource-bandwidth ratios

Figure 4. Software Architecture of AppRAN.

Together with the estimated bandwidth requirements, the controller computes the resource requirement of each application and determines how the resource should be divided among these applications. Resource allocation model of AppRAN will be discussed in Section III. The calculated resource allocation decisions are then disseminated to respective elements as resource *policies*. Instead of specifying resource allocation for each flow, AppRAN attempts to create resource policies on the application level while allowing the freedom of elements on how to fulfill these policies. In this way, AppRAN simplifies resource scheduling at the controller. We comment that this design is in line with the principles of software defined networking (SDN) architectures [16]. As the radio element has more accurate information of channel conditions as well as fine-grained power management and maturing modulation and coding technologies [17], allowing elements to optimize local resource allocation will further improve resource utility and enable heterogeneous network deployment. Discussion on such last-hop resource allocation, *i.e.,* from radio elements to respective mobile devices, is beyond the scope of this paper. Interested readers are referred to [17].

III. Optimal Resource Allocation in AppRAN

The resource allocation scheduler is the kernel of AppRAN, running at the logically central controller (see Figure 4), and periodically computes the resource distribution among the applications with respect to current network conditions as well as system configurations. In this section, we first develop an optimization framework for the resource allocation problem, analyze the hardness and then propose a fast algorithm which achieves a resource sharing decision provably close to optimal.

A. The Optimization Framework

1) Formulation: We target a radio access network with a set of I radio elements supporting a set of K applications in current configuration. Each radio element $i \in I$ has a resource amount B_i, representing the available radio resource at the element, abstracted from the three-dimensional resource grid.

Additionally, we denote the aggregate radio resource over all elements by B, *i.e.,* $B = \sum_{i \in I} B_i$. For isolation purpose, each application $k \in K$ reserves a minimum resource of L^k irrespective to traffic demands, while it can use up to M^k resource to improve its performance, where $L^k \leq M^k \leq B$. Similarly, the system also configures a lower bound l_i^k and a upper bound m_i^k for resource allocation to application k at element i to enable element-level isolation, accordingly, $L^k = \sum_{i \in I} l_i^k$ and $M^k = \sum_{i \in I} m_i^k$.

Let s_i^k be the amount of resource allocated to application k at element i, $i \in I, k \in K$. Now for each time period τ, we aim to maximize the overall resource allocation gain or utility while assuring that the resource used by each application is bounded according to preset configurations. Defining the utility function $u_i^k(\cdot)$, we formulate the following optimization problem:

$$\begin{aligned}
\max & \sum_{k \in K} \sum_{i \in I} u_i^k(s_i^k) \\
\text{s. t.} & \sum_{k \in K} s_i^k \leq B_i, \ \forall i \in I \\
& \sum_{i \in I} s_i^k \leq M^k, \ \forall k \in K \\
& \sum_{i \in I} s_i^k \geq L^k, \ \forall k \in K \\
\text{var.} \ & 0 \leq l_i^k \leq s_i^k \leq m_i^k, \ \forall i \in I, k \in K.
\end{aligned} \tag{1}$$

The first constraint indicates the resource limit at each radio element i. The second and third constraints impose the upper bound and lower bound of resource that can be allocated to each application k over the RAN, respectively. The per-element resource upper bound and lower bound for each application are imposed by the last constraint.

A similar optimization problem is formulated in [11], which attempts to maximize the aggregate utility of allocating network-wide radio resource proportionally to entities or mobile virtual network operators. However, they only consider the symmetric scenario by assuming that all base-stations (termed as radio elements in this paper) possess the same amount of resource, which prevents their framework from scaling to the complicated networking reality nowadays with heterogeneous wireless elements. On the contrary, our formulation directly addresses the amount of radio resource, allowing heterogeneity of elements. Providing the information of available resource at each element, these resource amounts are readily to be converted to percentages for implementation purposes. Therefore, problem (1) fundamentally differs from the model formulated in [11].

2) Utility Function and Demand Estimation: The utility function $u_i^k(\cdot)$ can be a linear function or any concave function following the law of diminishing marginal utility, representing the utility value of allocated resource to application k at element i. The following equations show examples of a linear function and a logarithmic function drawn from the proportional fairness principles defined in [18]:

$$u_i^k(s_i^k) = w_i^k \cdot d_i^k \cdot s_i^k, \qquad \text{or} \tag{2}$$

$$u_i^k(s_i^k) = w_i^k \cdot d_i^k \cdot \log(s_i^k), \tag{3}$$

where w_i^k is the utility weight, d_i^k is the resource demand for application k at element i in current period.

In conventional RAN sharing models, it is intractable to estimate resource demand d_i^k with respect to distinct entity, let alone to support differentiated QoS for different applications within one entity. For one thing, the network-level resource allocator has no information of the ownership of flows, *i.e.,* to which entity each flow belongs. This forces the system either to use the off-line, long-term estimation of "average" demands, or to employ deep packet inspection (DPI) to extract application-level information from flows. The former lacks accuracy, while the latter apparently introduces an intolerable computational overhead. For another, translating flow-level bandwidth demands to radio resource demands requires the information of modulation coding schemes (MCSs) selected for each flow transmission, which in turn relys on element-level details of users' channel conditions and MCS adaptation schemes [17].

In AppRAN, bandwidth demands of ongoing flows are irrespective of the entities they belong to, requiring no DPI operations. Mobile systems usually support Differentiated Services (DiffServ) [14] in their IP backbones for QoS management, *e.g.,* evolved packet system (EPS) bearers in LTE systems [4]. With DiffServ, AppRAN easily determines to which application a flow belongs by checking the QoS class identifier (QCI) attached to the flow. Moreover, together with the knowledge of channel conditions and scheduling algorithms, each radio element is ready to translate bandwidth demands of any application to resource demands. We define the average bandwidth-resource translating ratio for application k at radio element i as $p_i^k = \frac{resource\ to\ support\ application\ k\ with\ QoS}{bandwidth\ demand\ for\ application\ k}$, which is reported to the central controller for resource-demand estimation. Here, we note that such translating ratio might not reveal the "real" relation between bandwidth and resource demands in the cases with significant flow fluctuation, *e.g.,* the traffic demand of the user with the worst channel condition soars for the next time period τ, or the channel condition of a heavy-traffic user significantly changes. Yet we argue that our approach remains effective. This is because: (1) For a short time period of τ, it is less likely to have large fluctuation. Even with large fluctuation, the system only experiences suboptimal resource allocation for at most τ time; and (2) In AppRAN, we compute the resources allocated at each element to distinct applications. Therefore, the fluctuation can be mitigated or shaped by employing adaptive MCS schemes [19] at radio elements. That is, given an application and allocated resource, the element runs a second-phase resource allocation to distribute resource among flows with accurate channel state information.

3) Problem Hardness: If a linear function such as (2) is adopted as the utility function, problem (1) is a linear programming (LP) problem. Although exact solutions to LP problems are tractable, problem (1) has a prohibitively large size for any fast computation through modern LP solvers, *e.g.,* CPLEX [20]. To have a rough understanding, a production mobile system usually has $O(10^5)$ radio elements and supports $O(10^2)$ applications. Therefore, the rudimentary size of problem (1) is with $O(10^7)$ variables and $O(10^7)$ constraints (see the last constraint of problem (1)). If the utility is defined as a concave function, *e.g.,* using (3), problem (1) is a nonlinear programming problem [21], which has much higher computational complexity than the LP version and is infeasible for existing nonlinear solvers, *e.g.,* OPT++ [22], to compute a solution within reasonable time. Therefore, instead of pursuing exact solutions, a fast algorithm with approximate guarantees is more desirable.

B. The Approximate Algorithm

1) Main Procedure: We employ the Barrier method from [23] and solve problem (1) via an interior-point approach. For ease of presentation, let $\boldsymbol{s}$ be the vector of variables $\{s_i^k | i \in I, k \in K\}$. We define the logarithmic barrier function as

$$\begin{aligned}\phi(\boldsymbol{s}) =& \sum_{i\in I} \log(B_i - \sum_{k\in K} s_i^k) + \sum_{k\in K} \log(M^k - \sum_{i\in I} s_i^k) \\ &+ \sum_{k\in K} \log(\sum_{i\in I} s_i^k - L^k).\end{aligned} \tag{4}$$

We denote the objective of problem (1) by $u(\boldsymbol{s}) = \sum_{k\in K}\sum_{i\in I} u_i^k(s_i^k)$. We then introduce a multiplier t and consider the following problem:

$$\begin{aligned}&\max\ t\cdot u(\boldsymbol{s}) + \phi(\boldsymbol{s}) \\ &\text{s. t. } l_i^k \le s_i^k \le m_i^k,\ \forall i \in I, k \in K.\end{aligned} \tag{5}$$

The main procedure of our algorithm is listed in Algorithm 1. The procedure follows a typical route of the Barrier method, while we develop a tighter bound. Starting from a feasible point, it iteratively solves a sequence of problem (5) with increasing t till $t \ge (B + |K|)/\epsilon$ (to be discussed later). For simplicity, we set $\boldsymbol{s}_0 = \boldsymbol{l}$ to be the initial starting point, providing that $\boldsymbol{l}$ is a feasible solution to problem (1) under proper configurations, *i.e.,* $\sum_k l_i^k \le B_i, \forall i \in I$. Line 4 therein is called an inner loop for solving the optimization problem (5) with bounded variables. We refer readers to [23] for the details of the inner loop, *e.g.,* the Newton's method. Each solution found in line 4 is then used as a new starting point for the next iteration in the outer loop. Here, μ is a parameter involving a trade-off in the number of iterations of the inner and outer loops. Details on selecting μ can also be found in [23].

Algorithm 1: Barrier method for problem (1).

```
1: Start with an interior feasible point $\boldsymbol{s}_0$;
2: while $(B + |K|)/t > \epsilon$ $(\epsilon > 0)$ do
3:     $\boldsymbol{s} := \boldsymbol{s}_0,\ t := t_0$, where $t_0 > 0$;
4:     With starting point $\boldsymbol{x}$, solve (5) via a gradient-based
       method (e.g., Newton's method) and output the
       solution $\boldsymbol{x}^*$.
5:     $\boldsymbol{s} := \boldsymbol{s}^*,\ t := t \cdot \mu$, where $\mu > 1$;
6: end while
```

2) The Approximate Result: We now develop the theoretical basis for Algorithm 1. Applying the duality analysis, we have the following conclusion.

Lemma 1. *If problem (5) can be optimally solved, then we can find a solution to problem (1) that is at most ϵ−suboptimal, for any $\epsilon > 0$. In other words, let u^* be the optimal value of problem (1) and $\bar{\boldsymbol{s}}$ be the optimal solution to problem (5). By setting $t \ge (B + |K|)/\epsilon$, we have*

$$u^* \le u(\bar{\boldsymbol{s}}) + \epsilon.$$

Proof. According to the concavity of $t \cdot u(\boldsymbol{s}) + \phi(\boldsymbol{s})$, the following group of conditions are necessary and sufficient for an optimal solution to problem (5):

$$\begin{aligned}&\frac{\partial}{\partial s_i^k}(t\cdot u(\boldsymbol{s}) + \phi(\boldsymbol{s})) = 0, \text{ or} \\ &s_i^k = l_i^k \text{ and } \frac{\partial}{\partial s_i^k}(t\cdot u(\boldsymbol{s}) + \phi(\boldsymbol{s})) < 0, \text{ or} \\ &s_i^k = m_i^k \text{ and } \frac{\partial}{\partial s_i^k}(t\cdot u(\boldsymbol{s}) + \phi(\boldsymbol{s})) > 0.\end{aligned} \tag{6}$$

Define $p_i(\boldsymbol{s}) \triangleq \sum_{k\in K} s_i^k$, $q_k(\boldsymbol{s}) \triangleq \sum_{i\in I} s_i^k$. To expand the partial differential equations, we have

$$\begin{aligned}\frac{\partial}{\partial s_i^k}(t\cdot u(\boldsymbol{s}) + \phi(\boldsymbol{s})) &= t\cdot\frac{\partial u(\boldsymbol{s})}{\partial s_i^k} + \frac{-1}{B_i - p_i(\boldsymbol{s})} \\ &+ \frac{-1}{M^k - q_k(\boldsymbol{s})} + \frac{1}{q_k(\boldsymbol{s}) - L^k}.\end{aligned} \tag{7}$$

We now consider the dual problem of (1). The convex Lagrange dual function of problem (1) is listed as

$$\begin{aligned}\mathscr{L}(\boldsymbol{s}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}) &= -u(\boldsymbol{s}) + \sum_{i\in I} \lambda_i(p_i(\boldsymbol{s}) - B_i) \\ &+ \sum_{k\in K} \mu_k(q_k(\boldsymbol{s}) - M^k) + \sum_{k\in K} \nu_k(L^k - q_k(\boldsymbol{s})).\end{aligned} \tag{8}$$

The dual problem of problem (1) is then

$$D(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{l}\le\boldsymbol{s}\le\boldsymbol{m}} \mathscr{L}(\boldsymbol{s}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}).$$

Since $\mathscr{L}$ is a convex function, the following conditions are necessary and sufficient to minimize the dual function over $\boldsymbol{s} \in [\boldsymbol{l}, \boldsymbol{m}]$:

$$\begin{aligned}&\frac{\partial \mathscr{L}}{\partial s_i^k} = 0, \text{ or} \\ &s_i^k = l_i^k \text{ and } \frac{\partial \mathscr{L}}{\partial s_i^k} > 0, \text{ or} \\ &s_i^k = m_i^k \text{ and } \frac{\partial \mathscr{L}}{\partial s_i^k} < 0.\end{aligned} \tag{9}$$

Here we note that

$$\frac{\partial \mathscr{L}}{\partial s_i^k} = -\frac{\partial u(\boldsymbol{s})}{\partial s_i^k} + \lambda_i + \mu_k - \nu_k.$$

By setting

$$\begin{cases}\lambda_i = \dfrac{1}{t\cdot(B_i - p_i(\boldsymbol{s}))} \\ \mu_k = \max(0, \dfrac{1}{t\cdot(M^k - q_k(\boldsymbol{s}))} - \dfrac{1}{t\cdot(q_k(\boldsymbol{s}) - L^k)}) \\ \nu_k = \max(0, \dfrac{1}{t\cdot(q_k(\boldsymbol{s}) - L^k)} - \dfrac{1}{t\cdot(M^k - q_k(\boldsymbol{s}))})\end{cases}, \tag{10}$$

we, therefore, claim that the condition groups (6) and (9) are equivalent to each other. Correspondingly, if $\bar{\boldsymbol{s}}$ is the optimal

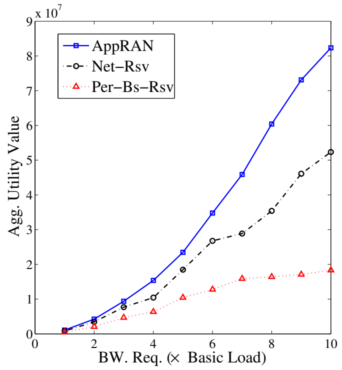


Figure 5. Utility with Linear Function.

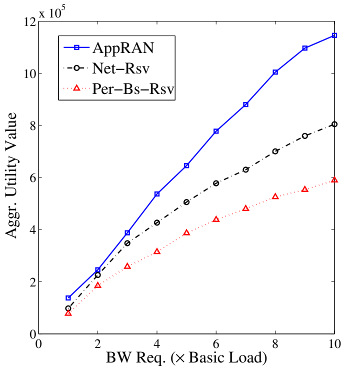


Figure 6. Utility with Log. Function.

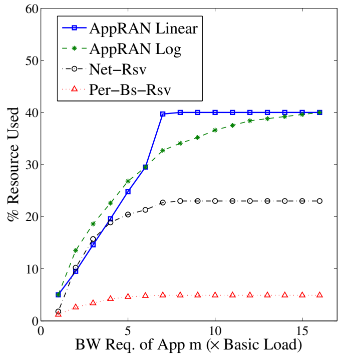


Figure 7. Resource Usage for Application m

solution to problem (5), we can derive non-negative dual multipliers $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}})$ that minimize $\mathscr{L}(\boldsymbol{s}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu})$. In other words, $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}})$ is dual feasible. According to the duality theory, we have

$$\begin{aligned} -u^* \geq D(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}}) &= \mathscr{L}(\bar{\boldsymbol{s}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}}) \\ &= -u(\bar{\boldsymbol{s}}) + \sum_{i \in I} \bar{\lambda}_i (p_i(\bar{\boldsymbol{s}}) - B_i) \\ &\quad + \sum_{k \in K} \bar{\mu}_k (q_k(\bar{\boldsymbol{s}}) - M^k) + \sum_{k \in K} \bar{\nu}_k (L^k - q_k(\bar{\boldsymbol{s}})) \\ &= -u(\bar{\boldsymbol{s}}) - \frac{B}{t} - \frac{1}{t} \sum_{k \in K} (1 - \min(\frac{M^k - \bar{q}_k}{\bar{q}_k - L^k}, \frac{\bar{q}_k - L^k}{M^k - \bar{q}_k})), \end{aligned} \tag{11}$$

in which $\bar{q}_k = q_k(\bar{\boldsymbol{s}})$. Accordingly,

$$\begin{aligned} u^* &\leq u(\bar{\boldsymbol{s}}) + \frac{B}{t} + \frac{1}{t} \sum_{k \in K} (1 - \min(\frac{M^k - \bar{q}_k}{\bar{q}_k - L^k}, \frac{\bar{q}_k - L^k}{M^k - \bar{q}_k})) \\ &\leq u(\bar{\boldsymbol{s}}) + \frac{B + |K|}{t} - \frac{\sum_k \min(\frac{M^k - \bar{q}_k}{\bar{q}_k - L^k}, \frac{\bar{q}_k - L^k}{M^k - \bar{q}_k})}{t} \\ &\leq u(\bar{\boldsymbol{s}}) + \frac{B + |K|}{t}. \end{aligned} \tag{12}$$

Therefore, setting $t \geq (B + |K|)/\epsilon$, we have $u^* \leq u(\bar{\boldsymbol{s}}) + \epsilon$. This completes the proof. □

The approximate result of Algorithm 1 is an instant result from Lemma 1. We state it in Theorem 1.

Theorem 1. *For any $\epsilon > 0$, Algorithm 1 obtains a solution to problem (1), which is at most $\epsilon-$suboptimal.*

IV. Simulation and Numerical Results

In this section, we study the AppRAN model through extensive computer simulations. We focus on the application-level resource allocation at the RAN gateway, while we assume that certain flow-level resource allocation schemes, *e.g.,* [17], are adopted by each radio element and radio resource therein is abstracted and represented as a non-negative real value using the technologies in [8,9]. The simulator is mainly developed in C++ with around 2000 lines of code.

A. Simulation Setup

We simulate a RAN system with $1,000$ radio elements shared by 20 entities. AppRAN defines 100 abstract applications with resource requirement factors uniformly selected within $[0.1, 2]$ per unit data rate (Mbps), representing respective QoS guarantees. We assume that all these applications are supported over all entities and radio elements to exclude the complexity of SLAs from our simulations. Given that the information of channel conditions is available to elements, the average resource-bandwidth multipliers are uniformly generated from $[1.0, 2.0]$, resulting in a resource-bandwidth ratio range of $[0.1, 4.0]$ (jointly determined by channel conditions and QoS requirements). The available resource at elements is abstracted as real values randomly generated from $[100.0, 300.0]$ with the mean value of 200.0 over all elements. This setting is to ensure that the logarithmic utility function results in a non-negative value in all cases, representing proportional resource capacities that can be arbitrarily scaled up/down over the system. Two most data-consuming applications, m and n, *e.g.,* video streaming and FTP file downloading, can use at least 5% and up to 40% of the aggregate resource, while other applications equally share the rest resource.

For comparison purpose, we align AppRAN with two alternative entity-oriented resource allocation schemes, termed as *Per-Base-station Reservation* (Per-Bs-Rsv) [10] and *Network Reservation* (Net-Rsv) [11], in which the utility is calculated over flows instead of entities as in the literature. In Net-Rsv, each entity reserves 2% of the aggregate resource and can use up to 10%, while in PerBs-Rsv, each entity reserves 5% resource at each radio element. For comparison fairness, the utility is calculated over flows for all three schemes to be comparable, since the accumulated utility either over applications or entities can be decomposed in the form of flows. The basic system load contains $5,000$ flows randomly generated from all applications across all radio elements, the bandwidth demands of which are selected from $0.1 \sim 1$ Mbps to ensure that a feasible resource assignment can be reached by all schemes. This load is then gradually increased by a controlled multiplier for different scenarios.

B. Utility Results

For simplicity, we set unit utility weights $w_i^k = 1 \; \forall i \in I$, $k \in K$. We use the linear function (2) and the logarithmic function (3) for utility calculations and show results in Figure 5 and Figure 6, respectively. In both scenarios, the system load increases step-by-step from 1 to 10 times of of the basic load. With a linear utility function, the quadratic-like utility-growing

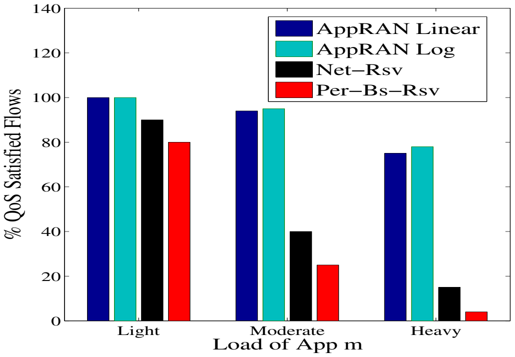


Figure 8. Comparsions of QoS Satisfied Flows.

curve of AppRAN in Figure 5 shows that AppRAN tends to allocate resource linearly to corresponding bandwidth demands when the system is under a low to moderate load. This growth is flattened with a logarithmic utility function (Figure 6), which also considers the fairness among applications. In both cases, AppRAN obtains a utility objective which significantly outperforms the Net-Rsv scheme (up to 40%) and the Per-Bs-Rsv scheme (up to 200%). This confirms the conclusion that resource reservation over entities immensely limits the RAN sharing performance with multiple QoS-differentiated applications.

C. QoS Results

In this simulation, we study the QoS performance of different schemes by gradually increasing the bandwidth demands of application m, one of the most data-consuming applications. Beyond the basic load setup in Section IV-A, we add $2,000$ extra flows of application m from 5 entities to randomly selected 200 radio elements with a mean basic bandwidth demand of 1.0 Mbps. The load of application m is then iteratively increased from 1 to 15 times of the basic load, while the demands of rest flows keep unchanged. Figure 7 shows the resource consumption of application m. It indicates that AppRAN effectively adapts resource allocation for data-consuming applications as the traffic demands increase. In contrast, as constrained by per-entity resource limits, both Net-Rsv and Per-Bs-Rsv result in significant resource under-utilization irrespective of idle resource in the system. In AppRAN, we also observe that with the linear utility function, the resource usage grows more aggressively to its resource upper bound.

In Figure 8, we show the number of QoS satisfied flows in each case. Here, light, moderate and heavy loads correspond to 1, 10 and 15 times of the basic load of application m. We can see from Figure 8 that AppRAN achieves similar QoS performance with the linear or logarithmic utility function. However, with entity-oriented resource reservation, Net-Rsv cannot even fully support all flows with the least load and the performance deteriorates further as the load increases. Likewise, Per-Bs-Rsv produces the worst performance due to its strict entity-oriented resource constraints.

V. Conclusion

In this paper, we propose AppRAN, an application-oriented framework for sharing RAN resource. AppRAN defines a serial of abstract applications according to the differentiated QoS requirements and provides service level agreements by letting entities map their concrete applications to corresponding abstract applications. AppRAN centrally optimizes resource distribution among applications at each element, while the decisions on allocating resource to flows are determined distributively at each element with real-time channel conditions. Flows inside AppRAN are identified only by QCIs or bearers, irrespective to their belonging entities. Therefore, the design of AppRAN is in line with the principles of SDN and enables good network abstraction. We also propose a fast algorithm to optimize the resource allocation in AppRAN and prove its approximate guarantee. The simulation results demonstrate significant performance improvement over entity-oriented schemes in terms of aggregate utility and QoS satisfaction.

References

[1] "GSMA report: Mobile infrastructure sharing," http://www.gsma.com/publicpolicy/mobile-infrastructure-sharing-report/, November 2008.

[2] Visiongain, "Mobile network sharing report 2010-2015, development, analysis and forecasts," August 2010.

[3] A. R. Mishra, *Fundamentals of Cellular Network Planning and Optimisation: 2G/2.5 G/3G ... Evolution to 4G*. John Wiley & Sons, 2004.

[4] S. Sesia, I. Toufik, and M. Baker, *LTE: The UMTS Long Term Evolution*. Wiley Online Library, 2009.

[5] Telcordia, "With the right support, MVNOs can enrich network operators with innovation, differentiation, and market share," White Paper, 2012.

[6] Visiongain, "Mobile virtual network operator (MVNOs) market forecast 2014-2019," September 2014.

[7] M. Hoffmann and M. Staufer, "Network virtualization for future mobile networks: General architecture and applications," in *IEEE International Conference on Communications Workshops (ICC)*, 2011, pp. 1–5.

[8] S. Katti and L. E. Li, "Radiovisor: A slicing plane for radio access networks," *Open Networking Summit 2014 (ONS 2014)*, 2014.

[9] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proceedings of 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, 2013, pp. 25–30.

[10] R. Kikku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1333–1346, 2012.

[11] R. Mahindra, A. Khojastepour, H. Zhang, and S. Rangarajan, "Network-wide radio access network sharing in cellular networks," in *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2013.

[12] F. Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*. Cambridge University Press, 2009.

[13] IXIA, "Quality of service and policy management in mobile data networks," December 2013.

[14] K. Nichols *et al.*, "RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," 1998.

[15] ADVA *et al.*, "Horizon 2020 advanced 5G network infrastructure for future Internet PPP, draft version 2.1," 2013.

[16] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, 2008, pp. 69–74.

[17] Z. Han and K. R. Liu, *Resource Allocation for Wireless Networks*. Cambridge University Press, 2008.

[18] G. Tychogiorgos, A. Gkelias, and K. K. Leung, "Utility-proportional fairness in wireless networks," in *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2012, pp. 839–844.

[19] S. T. Chung and A. J. Goldsmith, "Degrees of freedom in adaptive modulation: A unified view," *IEEE Transactions on Communications*, vol. 49, no. 9, pp. 1561–1571, 2001.

[20] IBM Inc., "IBM ILOG CPLEX Optimizer," http://www.ibm.com/software/commerce/optimization/cplex-optimizer/.

[21] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, 2013.

[22] J. Meza, P. Hough, and P. Williams, "OPT++: An object-oriented nonlinear optimization library, 2004," http://software.sandia.gov/opt++/.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.