R-Cloud: A Cloud Framework for Enabling Radio-as-a-Service over a Wireless Substrate

Chenfei Gao, Gozde Ozcan, Jian Tang, Mustafa Cenk Gursoy and Weiyi Zhang

***Abstract*—Inspired by the success of use of Virtual Machines (VMs) in cloud computing, virtualization has been introduced to wireless networking recently, enabling support for multiple Mobile Virtual Network Operators (MVNOs) via isolated slices over a shared wireless substrate. In this paper, we present design, implementation and evaluation of a novel cloud framework, R-Cloud, to enable radio resources at Base Stations (BSs) to be effectively allocated to multiple MVNOs as a service, which is referred to as *Radio-as-a-Service (RaaS)*. R-Cloud employs a hybrid two-level control framework to enable coarse-grained and fine-grained resource allocation at the cloud and BS levels respectively. Specifically, R-Cloud not only coordinates resource allocation among BSs, MVNOs and mobile users across a Radio Access Network (RAN) and enables performance isolation by optimizing resource sharing and user association using an LP-rounding based algorithm at the cloud level; but also effectively schedule transmissions among multiple users at a BS using an optimal scheduling policy. We implemented R-Cloud over a wireless network testbed with software-defined radios. It has been shown by extensive experimental and simulation results that R-Cloud can achieve effective RaaS over wireless networks and the proposed resource allocation algorithms outperform widely-used baseline solutions.**

***Index Terms*–Cloud Computing, Wireless Virtualization, Radio Resource Management, Radio Access Network**

I. Introduction

Recently, we have witnessed dramatic growth of wireless traffic as a result of the transition from voice to video/data domination as well as increasing popularity of smartphones, tablets and Internet of Things (IoT). To accommodate increasing wireless traffic, mobile network operators are contingent upon deploying micro-cells to improve network capacity and coverage. In a traditional Radio Access Network (RAN), each Base Station (BS) makes decisions independently with loose coordination. This approach works well for a RAN with macro-cells, in which BSs cover relatively large and non-overlapping regions and a wireless terminal usually can be covered by one and only one BS. However, current RAN consists of many micro-cells, in which an area is densely covered by many micro-BSs. In this case, traditional distributed control at BSs may not work well due to lack of a global view of the whole wireless network and optimized control over radio resources.

Chenfei Gao, Gozde Ozcan, Jian Tang and Mustafa Cenk Gursoy are all with Department of Electrical Engineering and Computer Science at Syracuse University, Syracuse, NY 13244. Email: {cgao03, gozcan, jtang02, mcgursoy}@syr.edu. Weiyi Zhang is with AT&T Labs Research, Middletown, NJ 07748. Email: wzhang@ieee.org This work was done when Weiyi Zhang was with North Dakota State University. This research was supported by NSF under grant #1443966. The information reported here does not reflect the position or the policy of the federal government.

Cloud computing has emerged as a *de facto* computing model, enabling software, infrastructure, resources and data to be used as services over the network in an on-demand manner. Virtualization has been widely used in cloud computing to achieve resource sharing and performance isolation among multiple tenants. For example, in a cloud, Virtual Machines (VMs) can be created to host applications/services and a tenant can rent multiple VMs from the cloud service provider to run services for its own users. Inspired by the success of use of VMs in cloud computing, virtualization has been introduced to wireless networking [16] recently, enabling support for multiple Mobile Virtual Network Operators (MVNOs) via isolated slices over a shared wireless substrate. An MVNO is a wireless service provider that does not own the physical wireless network. With wireless network virtualization, a mobile network operator (such as AT&T and Verizon) can lease its radio resources at BSs (as a service) to an MVNO such that it can offer services to its own users. This is similar to the Infrastructure-as-a-Service (IaaS) model in cloud computing, in which a tenant rents VMs from the cloud service provider (such as Amazon and Google) to support its own services or users.

Wireless network virtualization is quite challenging due to time-varying and hard-to-predict wireless channels and link states. Hence, the corresponding research is still in its infancy. Moreover, most existing works [2], [3], [12], [16], [17], [20], [27], [28] targeted at a single Base Station (BS) rather than a RAN. In this paper, we propose a novel cloud framework, R-Cloud, to enable radio resources at BSs to be effectively allocated to multiple MVNOs as a service, which is referred to as *Radio-as-a-Service (RaaS)*. We aim to leverage R-Cloud for not only effectively sharing resources among MVNOs at a BS but also coordinating resource allocation among BSs, MVNOs and mobile users across a RAN via optimized resource sharing and user association. With R-Cloud, BSs in a RAN will work as a "big radio" to serve its users in the best way, as envisioned in [11]. Moreover, we aim to design a hybrid framework that seamlessly combines both centralized control at the cloud, and distributed control and data processing at BSs. This differentiates our work from rather extreme cloud-based solutions [4], [7], [21], [26] that use data centers to handle almost everything, which require links between BSs and data centers to have a very high capacity and extremely low latency. In summary, R-Cloud has the following desirable features:

- *Hybrid control for resource allocation*: R-Cloud employs a hybrid two-level control framework to enable coarse-

978-1-5090-3281-5/16/$31.00 ©2016 IEEE

grained and fine-grained resource allocation at cloud and BS levels respectively, which leads to a good tradeoff between distributed and centralized control.

- *Effective resource sharing at the cloud level (Section III-B)*: In R-Cloud, radio resources at BSs across the whole RAN are shared by multiple MVNOs using an effective algorithm that optimizes resource sharing and user association according to both Service Level Agreements (SLAs) of MVNOs and runtime states.
- *Optimal transmission scheduling at the BS level (Section III-C)*: R-Cloud uses an optimal policy for transmission scheduling at a BS, which maximizes the total effective capacity of users while satisfying the statistical Quality of Service (QoS) requirements according to the resource sharing solution provided at the cloud level.
- *Performance isolation*: In R-Cloud, changes in an MVNO (such as the number of wireless users, their traffic load, etc) do not affect wireless users of other MVNOs.

The rest of the paper is organized as follows: We discuss related work in Section II. We present the architecture of R-Cloud, describe its two-level control framework and discuss implementation details in Section III. Then we discuss testbed setup, and present and analyze experimental and simulation results in Section IV. We conclude the paper in Section V.

II. Related work

We discuss related works on both wireless network virtualization and cloud-based RAN.

Wireless Network Virtualization: Virtualized systems and virtualization methods have been proposed and implemented over various wireless substrates, such as WiMAX [3], [16], [17], [19], WiFi [2], [12] and LTE [27], [28]. In an early work [3], the authors presented an architecture that implements virtualization of radio resources to achieve isolation among multiple MVNOs and uses an algorithm for weighted fair sharing among multiple slices. In [16], the authors implemented both uplink and downlink virtualization over a WiMAX substrate. They introduced a slice scheduler that allows existence of slices with bandwidth-based and resource-based reservation simultaneously and includes a generic framework enabling customized flow scheduling within a BS on a per-slice basis. In another work [17], they extended their early works to RAN sharing by designing and implementing CellSlice, a system for slicing wireless resources in a RAN. The paper [19] presented the design and implementation of NetShare, a network-wide radio resource management framework that provides effective RAN sharing. NetShare introduces a two-level scheduler split between the mobile gateway and the cellular BSs to effectively manage and allocate the wireless resources of the RAN composed of multiple BSs among multiple MVNOs.

Wireless network virtualization has also been addressed in the context of WiFi. The SplitAP architecture was presented in [2], which addresses the problem of sharing uplink airtime across groups of users. Also, they designed and evaluated the LPFC and LPFC+ algorithms for group uplink airtime control using SplitAP on commercial off-the-shelf hardware. The authors of [20] proposed an airtime-based resource control technique based on the enhanced IEEE 802.11e EDCA for WLAN virtualization. Guo *et. al.* [12] developed a system, ViFi, which guarantees proportional fair share of channel access time at group level and isolates traffic between user groups. Some related works, such as [27] and [28], targeted at LTE network virtualization. The authors of [27] presented a framework, which virtualizes LTE systems so that multiple operators can share the same physical resources while being able to stay isolated from each other. They aimed to exploit the advantages that can be obtained from virtualizing the air interface. Two possible gains were explored: spectrum multiplexing and multi-user diversity. In [28], the authors first presented an analytical model of FTP transmissions in a virtualized LTE system. Then, they presented an extended multi-party spectrum sharing model and analyzed spectrum budget estimation based on the characteristics of real-time services.

Cloud-based Wireless Networking: China Mobile proposed a Cloud-RAN (C-RAN) architecture [7], which introduces a resource pool to process wireless baseband data in a cloud and replaces existing BSs with just the antennas and a few other active RF components. Solutions with similar architectures have been presented in [4], [21]. In [4], Bhaumik *et. al.* designed a framework, CloudIQ, to achieve two objectives: 1) partitioning the set of BSs into groups that are simultaneously processed on a shared homogeneous compute platform for a given statistical guarantee, and 2) scheduling the set of BSs allocated to a platform in order to meet their real-time processing requirements. In [21], the authors proposed FluidNet, a scalable and light-weight framework for realizing the full potential of C-RAN. FluidNet deploys a logically re-configurable front-haul to apply appropriate transmission strategies in different parts of the network and hence cater effectively to both heterogeneous user profiles and dynamic traffic load patterns. However, the authors of [11] argued that pushing all data processing and control to a central entity imposes huge demands on bandwidth and latency required on the backhaul. They suggested that data processing and latency-sensitive decision making continue to be handled by the BS and other control plane functionalities could be moved to the cloud. They proposed SoftRAN, a software-defined centralized control plane for RANs that abstracts all BSs in a local geographical area as a virtual big BS consisting of a central controller and radio elements. In [26], Yang *et. al.* proposed OpenRAN, an architecture for software-defined RAN via virtualization, which achieves virtualization and programmability vertically, and benefits convergence of heterogeneous network horizontally.

Another line of research [15], [18] aimed to exploit how to leverage emerging Software Defined Networking (SDN) for enhancing cellular networks, particularly core networks. For example, Jin *et. al.* in [15] presented SoftCell, a scalable and SDN-based architecture that supports fine-grained policies for mobile devices in cellular core networks, using commodity switches and servers.

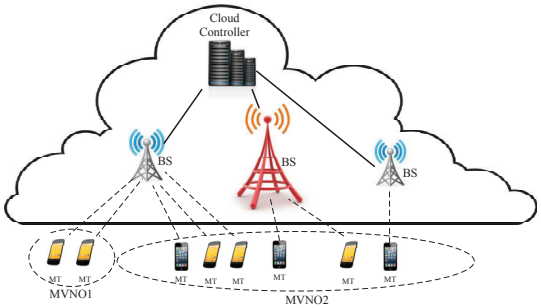


Fig. 1: The architecture of R-Cloud

The differences between R-Cloud and the above related works are summarized as follows: 1) Unlike some related works focusing on virtualization at a single BS [2], [3], [12], [16], [17], [20], [27], [28], we target at virtualization and resource sharing across a RAN with multiple BSs. 2) Unlike those solutions that pushed all data processing and control to the cloud [4], [7], [21], [26], we proposed a hybrid solution that handles data processing and part of control still at BSs while using the cloud to control resource allocation across the RAN. 3) The proposed R-Cloud makes decisions on user association with consideration for both SLAs and runtime states, which is different from those works that assume user association is known in advance [2], [3], [12], [16], [17], [19], [20], [27], [28]. 4) Unlike some related works that lack real implementation and experiments [11], [20], [26]–[28], we validate and evaluate R-Cloud on a wireless network testbed with software-defined radios. 5) Our main focus is not SDN or core networks, which differentiates our work from [15], [18]. 6) Different from a closely related work [11] that only introduced the "big base station" concept, we propose a comprehensive solution to realize RaaS and conducted extensive experiments and simulation to validate and evaluate it.

III. Design and Implementation of R-Cloud

In this section, we present the architecture of R-Cloud, introduce its two-level control framework and discuss the implementation in details.

A. System Architecture

As illustrated in Figure 1, a RaaS cloud (R-Cloud) consists of one or multiple cloud controllers and a set of BSs in the field. A centralized cloud controller can be placed in a data center, which is connected with BSs with high-capacity backhaul links (such as optical fibers). The controller supervises MVNOs and their users, and provisions radios resources to MVNOs dynamically according to runtime states (such as user demands, data rate of a BS, channel/link states, etc), and Service Level Agreements (SLAs) of MVNOs. In our design, we come up with a two-level (cloud-level and BS-level) control framework for resource allocation (Figure 2), which consists of the following modules:

- Network Monitor (on the cloud controller): it collects states of each BS (such as channel/link states, data rate, associated users, etc) from the BS manager running at the BS.
- SLA Manager (on the cloud controller): It maintains SLA information (Section III-B) of all MVNOs.
- Cloud-level Optimizer (on the cloud controller) (Section III-B): It applies an optimization algorithm to determine user association and resource sharing among MVNOs at each BS according to SLAs and runtime states.
- BS-level Scheduler (on the BS) (Section III-C): It applies an optimal policy for transmission scheduling among multiple users at a BS according to the resource sharing solution provided by the cloud-level optimizer.
- BS Manager (on the BS): It actually operates the BS to perform the scheduling policy provided by the BS-level scheduler, and it also collects states of the BS and reports them to the network monitor on the cloud controller.

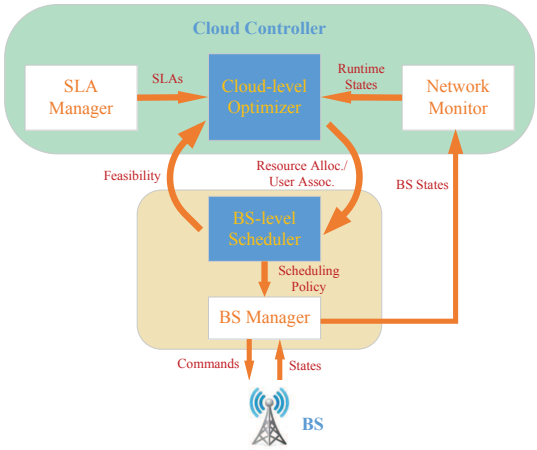


Fig. 2: The hybrid two-level control framework in R-Cloud

Control at the cloud level (Section III-B) is coarse-grained since the cloud controller only determines how the resources are shared among multiple MVNOs at each BS (i.e., resource sharing solution) and how to associate wireless users with BSs (i.e., user association) while leaving the actual resource allocation (such as transmission scheduling) at a BS for each wireless user to the BS-level scheduler (Section III-C), which is referred to as fine-grained control. Moreover, in the framework, the cloud-level optimizer and the BS-level scheduler can form a closed-loop. The cloud-level optimizer provides high-level (cloud-level) guidance for the BS-level scheduler to make decisions on resource allocation for individual users, which, in turn, provides feedback (e.g, if a feasible resource allocation can be found) for the cloud-level optimizer to adjust its solution. As mentioned above, our design of R-Cloud leads to the following desirable features: 1) hybrid control for resource allocation: this is due to coarse-grained and fine-grained control at cloud and BS level respectively; 2) Effective resource sharing at the cloud level: this is achieved by an effective LP-rounding based algorithm (Section III-B); 3) optimal transmission scheduling at a BS: this is achieved by an

optimal scheduling policy (Section III-C); and 4) performance isolation: this is achieved via resource allocation at the cloud-level. Next, we will explain how these are achieved in details.

B. Cloud-Level Optimizer

As shown in Figure 2, the cloud-level optimizer is a key component of the cloud controller, which optimizes resource sharing among MVNOs according to both their SLAs (given in advance) and runtime states. The major notations used in this section are summarized in Table I.

TABLE I: Major Notations

Notation	Description
b and B	BS b and the set of BSs
$\bar{r_b}$	Runtime average data rate of BS b
$r_{u,v}$	The traffic demand (i.e., min requested data rate) of user u of MVNO v
t_u	The fraction of resources allocated to user u at his/her associated BS
u and U	User u and the set of users
U_v	The set of users of MVNO v
v and V	MVNO v and the set of MVNOs
w_u	The weight of user u
$x_{b,v}$	The fraction of resources at BS b allocated to MVNO v
$y_{u,v,b}$	Binary user association variable for user u of MVNO v at BS b
γ_v	Resource reservation ratio of MVNO v
ϵ_v	Runtime violation ratio of MVNO v

We consider a RAN with a set B of BSs and a set V of MVNOs. Each MVNO $v \in V$ has its own set U_v of contracted users. Similar as the resource provisioning model in [19], the SLA of an MVNO v is given by the resource reservation ratio γ_v, which is the fraction of total resources that are reserved in advance for MVNO v. However, since wireless channel and link data rate are time-varying, reserved resources may not be sufficient for supporting traffic demands of mobile users of a MVNO at runtime. Hence, the SLA includes another parameter, runtime violation ratio ϵ_v, which is a threshold (in percentage) specifying how much an MVNO can tolerate at a BS in case of insufficient resources at runtime. An MVNO is supposed to pay more if it desires to have a larger γ_v and smaller ϵ_v. The cloud-level optimization aims to minimize the total unsatisfied traffic demands in the network while ensuring the SLA requirements are met for each MVNO. Unlike related work [19] in which user association is assumed to be given, we try to optimize it in our problem since we believe significant savings can be achieved by strategically arranging user association across the RAN. We formally present our problem formulation in the following.

MILP-Cloud:

- Resource allocation ratio variables:
  $\mathbf{x} = \{x_{b,v} \in [0,1] | b \in B, v \in V\}$: $x_{b,v}$ gives the fraction of resource at BS b allocated to MVNO v.
- User association variables:
  $\mathbf{y} = \{y_{u,v,b} = \{0,1\} | u \in U_v, v \in V, b \in B\}$: $y_{u,v,b} = 1$ if user $u \in U_v$ is associated with BS b; $y_{u,v,b} = 0$, otherwise.

$$\min_{<\mathbf{x},\mathbf{y}>} \sum_{b \in B} \sum_{v \in V} \max(0, \sum_{u \in U_v} r_{u,v} y_{u,v,b} - x_{b,v} \bar{r_b}) \quad (1)$$

Subject to:

$$\sum_{v \in V} x_{b,v} \leq 1, \forall b \in B; \quad (2)$$

$$\sum_{b \in B} y_{u,v,b} = 1, \forall u \in U_v, \forall v \in V; \quad (3)$$

$$\frac{\sum_{b \in B} x_{b,v} c_b}{\sum_{b \in B} c_b} \geq \gamma_v, \forall v \in V; \quad (4)$$

$$\frac{\sum_{u \in U_v} r_{u,v} y_{u,v,b} - x_{b,v} \bar{r_b}}{\sum_{u \in U_v} r_{u,v} y_{u,v,b}} \leq \epsilon_v, \forall b \in B, \forall v \in V. \quad (5)$$

The objective (1) is to minimize the total unsatisfied traffic demand in the network. Note that for MVNO v at BS b, if the corresponding unsatisfied traffic demand is 0, v's traffic demand can be fully satisfied at runtime. In the implementation, we introduce new continuous variables $t_{b,v}$:

$$t_{b,v} = \max(0, \sum_{u \in U_v} r_{u,v} y_{u,v,b} - x_{b,v} \bar{r_b}); \forall b \in B, \forall v \in V. \quad (6)$$

Then the objective function can be replaced by a linear function and two linear constraints:

$$\min_{<\mathbf{x},\mathbf{y},\mathbf{t}>} \sum_{b \in B} \sum_{v \in V} t_{b,v} \quad (7)$$

$$t_{b,v} \geq 0, \forall b \in B, \forall v \in V; \quad (8)$$

$$t_{b,v} \geq \sum_{u \in U_v} r_{u,v} y_{u,v,b} - x_{b,v} \bar{r_b}, \forall b \in B, \forall v \in V; \quad (9)$$

Constraints (2) ensure that the sum of fraction of resources allocated to every MVNO at a BS does not exceed 1. Constraints (3) make sure that each user can be served by one and only one BS. Constraints (4) and (5) correspond to the SLA requirements. Specifically, Constraints (4) make sure that the total resources allocated to each MVNO v ($\forall v \in V$) across the whole network is no less than the resource reservation ratio γ_v specified in its SLA. Note that c_b is the resource capacity of BS b, which can be set to 1 for a RAN with homogeneous BSs; or the ratio between the resource capacity of BS b and that of the baseline BS for the heterogeneous case. We set them to 1 in our experiments and simulation. Moreover, with constraints (5), the fraction of unsatisfied traffic demands of a MVNO at each BS is guaranteed to be no larger than the given threshold ϵ_v. Note that performance isolation is achieved via these constraints since they can ensure an MVNO to obtain certain portion of resources it deserves to have, which will not be affected by other MVNOs.

Here, the traffic demand of each user $u \in U_v$ of MVNO v ($r_{u,v}$), the runtime average data rate of each BS b ($\bar{r_b}$), resource reservation ratio of each MVNO v (γ_v) and runtime violation ratio of each MVNO v (ϵ_v) are given as input. Note that $< r_{u,v} >$ are the minimum data rates users request to have. And, $< \bar{r_b} >$ can be estimated or predicted (using a machine learning algorithm) based on both historical data and runtime states. The estimation or prediction should be done periodically to update their values. In addition, in practise, it is not beneficial to associate a user with a BS that is far away since signal strength may be very weak. So a distance threshold or certain rules can be set according to estimation (e.g., estimation based on some wireless signal propagation model) such that a subset of BSs can be excluded from consideration by setting the corresponding user association variables $< y_{u,v,b} >:= 0$.

MILP-Cloud is an MILP problem, which may take exponentially long time to solve, especially for large cases. Hence, we present a polynomial time heuristic algorithm to solve it. We seek to jointly determine user association $< y_{u,v,b} >$ and resource allocation ratios $< x_{b,v} >$ by an LP-rounding based algorithm. We denote Relaxed-LP-Cloud($\cdot$) as the LP relaxation of MILP-Cloud in which all the binary variables $< y_{u,v,b} >$ are relaxed to continuous variables $< y'_{u,v,b} > \in [0, 1]$. We solve Relaxed-LP-Cloud iteratively until all user associations are determined. In each iteration, we round up k largest $< y'_{u,v,b} >$ to 1 and set corresponding $< y_{u,v,b} >$ to 1. Meanwhile, we round down those $< y'_{u,v,b} >$ (with conflicts) to 0 and set corresponding $< y_{u,v,b} >$ to 0 to avoid violating Constraints (3). Once all values of $< y_{u,v,b} >$ have been determined, resource allocation ratios $< x_{b,v} >$ can be calculated accordingly by solving the last Relaxed-LP-Cloud. The algorithm is formally presented as Algorithm 1. Cloud-level Optimizer will run the algorithm periodically to update user association and resource allocation ratios according to runtime states.

Algorithm 1: Cloud-Level Optimizer

Input : V, B, $U =< U_v >$, $R =< r_{u,v} >$, $\bar{R} =< \bar{r_b} >$, $\Upsilon =< \epsilon_v >$, $\Gamma =< \gamma_v >$, k
Output: $< x_{b,v} >$, $< y_{u,v,b} >$

1 **while** *(1)* **do**
2 $< \mathbf{x}, \mathbf{y}', \mathbf{t} >:=$ Relaxed-LP-Cloud($R, \bar{R}, \Upsilon, \Gamma, \mathbf{y}'$);
3 **if** *(all values of $< y_{u,v,b} >$ are determined)* **then**
4 break;
5 **else**
6 **for** *($\forall y'_{u,v,b} \in L_k$, where L_k includes k largest such variables)* **do**
7 $y'_{u,v,b} := 1$;
8 $y'_{u,v,b'} := 0, \forall b' \neq b$;
9 $y_{u,v,b} := 1$;
10 $y_{u,v,b'} := 0, \forall b' \neq b$;
11 **return** $< x_{b,v} >$ and $< y_{u,v,b} >$;

In Algorithm 1, we solve Relaxed-LP-Cloud iteratively (lines 1–10). In each iteration of the while loop, we first solve Relaxed-LP-Cloud (line 2) then check the values of $< y_{u,v,b} >$ (line 3). If all values of $< y_{u,v,b} >$ are determined, it breaks out the loop (line 4) and returns the results (line 11). Otherwise, we conduct the rounding procedure (lines 6–10)) as described above. k is an input parameter that is used to tradeoff performance and running time. The lower the value of k, the better solution we will more likely obtain, however, more Relax-LP-Cloud need to be solved, which leads to longer running time. In our implementation, we set $k = 1$ in our experiments and $k = 5$ in our simulation. The time complexity of Algorithm 1 is dominated by the while loop, which takes $O((\frac{N}{k}) \cdot T_{\text{Relaxed-LP-Cloud}})$ time, where N is the total number of users in the network and $T_{\text{Relaxed-LP-Cloud}}$ is the time for solving the LP. In practise, we found that the LP can be solved very quickly, even for fairly large cases in our simulation.

C. BS-Level Scheduler

Based upon user association $< y_{u,v,b} >$ provided by the cloud optimizer, each BS finds out the set of users it needs to serve. Moreover, the BS-level scheduler schedules transmissions of its associated users using a policy guided by resource allocation ratios $< x_{b,v} >$ (given by the cloud optimizer), which is discussed next.

Before introducing BS-level scheduling problem which aims at maximizing the sum of effective capacities of the associated users, we first give a brief overview of the effective capacity. By applying the theory of large deviations, it was shown in [6] that for a dynamic queuing system where arrival and service processes are stationary and ergodic, the queue length process $Q(t)$ converges in distribution to a random variable $Q(\infty)$ such that

$$\lim_{Q_{\text{th}} \to \infty} -\frac{\log P(Q(\infty) \geq Q_{\text{th}})}{Q_{\text{th}}} = \theta, \tag{10}$$

where Q_{th} denotes the queue length threshold and θ is the so called QoS exponent. Sufficiently large Q_{th} yields the following approximation for the buffer overflow probability:

$$P(Q(\infty) \geq Q_{\text{th}}) \approx \mathrm{e}^{-\theta Q_{\text{th}}}. \tag{11}$$

From the above approximation, we can see that θ controls the exponential decay rate of the buffer overflow probability. In particular, larger θ implies that the system can support stringent QoS requirements since faster decay rate is imposed, which leads to smaller overflow probability. On the other hand, smaller values of θ indicate slower decay rate, which implies looser QoS requirements. Asymptotically, when $\theta \to \infty$, the system cannot tolerate any delay. On the other hand, $\theta \to 0$ implies that the system can tolerate long delays.

QoS requirements are critical considerations for delay sensitive applications, such as video conferencing and online gaming, in order to provide a satisfactory user experience. However, since wireless networks have randomly changing channel conditions due to fading, it is challenging to satisfy deterministic delay QoS constraints, and hence constraints on the delay-bound violation probability or the buffer overflow

probability are considered as statistical QoS requirements. In order to facilitate the design of communication systems operating under such QoS constrains, it is required that QoS provisioning is incorporated into wireless channel models. In this regard, the theory of the effective capacity is developed to analyze the performance of wireless systems under statistical QoS constraints. More specifically, the effective capacity characterizes the maximum constant arrival rate that can be supported by time-varying wireless transmissions while satisfying the statistical buffer constraint in (10). The effective capacity for a given QoS exponent θ is formulated as [6], [25]

$$C_E(\theta) = -\lim_{t\to\infty} \frac{1}{\theta t} \log_e \mathbb{E}\{e^{-\theta \sum_{j=1}^{t} R[j]}\}. \quad (12)$$

Above, $\{R[j]\}$ is the discrete-time stationary and ergodic stochastic service process. If the service-rate sequence $\{R[j]\}$ is uncorrelated, the effective capacity is simplified in block-fading channels as

$$C_E(\theta) = -\frac{1}{\theta} \log_e(\mathbb{E}\{e^{-\theta R[j]}\}). \quad (13)$$

Hence, by using the above characterization of the effective capacity, we propose an efficient QoS-driven scheduling among users while satisfying the statistical QoS constraints of each BS in the form of limitations on the buffer overflow probability. Respectively, the BS-level scheduling problem which aims at maximizing the weighted sum of associated users' effective capacities can be formulated as follows:

CP-BS:

$$\max_{<t_u>} \sum_{u\in U} w_u C_E(\theta_u, R_u, t_u) \quad (14)$$

subject to

$$\sum_{u\in U_{b,v}} t_u \le x_{b,v}, \quad \forall b \in B, \forall v \in V; \quad (15)$$

$$L_u \le t_u \le H_u, \quad \forall u \in U_{b,v}. \quad (16)$$

$C_E(\theta_u, R_u, t_u)$ in the objective function (14) is defined as

$$C_E(\theta_u, R_u, t_u) = -\frac{1}{\theta_u} \log\left(\mathbb{E}\{e^{-\theta_u t_u R_u}\}\right). \quad (17)$$

In this formulation, t_u is the fraction of time allocated to user u and R_u is the transmission rate of user u, $\mathbb{E}\{.\}$ denotes the expectation operation. Also, w_u in (14) denotes the weight assigned to user u and is computed by $\beta_u r_{u,v}$, where β_u is related to the user traffic priority and $r_{u,v}$ is the demand of user u belonging to MVNO$_v$. Moreover, $x_{b,v}$ in (15) is the maximum total fraction of resources allocated to the users associated with MVNO v at BS b, which is given by the cloud optimizer. Finally, L_u and H_u given in (16) represent the lower and upper bound on t_u, respectively. Different users have different L_u and H_u, which are contingent upon the type of user traffic.

Proposition 1 (CP-BS)**.** *The BS-level scheduling problem is a concave optimization problem.*

Proof. First, it can be easily verified that $f(t_u) = e^{-\theta_u t_u R_u}$ is a log-convex and non-increasing function in t_u for any fixed θ_u and R_u, which takes non-negative values. Since expectation preserves log-convexity, $\mathbb{E}\{e^{-\theta_u t_u R_u}\}$ is also log-convex [5]. This implies that $\log(\mathbb{E}\{e^{-\theta_u t_u R_u}\})$ is a convex function of t_u. Since the negative of a convex function is concave, it follows that the effective capacity of each user is a concave function [5]. As a result, the objective function in (14) being a nonnegative weighted sum of concave functions, is itself concave. Hence, the optimization problem involves the maximization of a concave function subject to affine constraints and the optimal fractions of time $< t_u^{\text{opt}} >$ can be obtained by employing a convex optimization tool. □

The optimal fractions of time $< t_u^{\text{opt}} >$ form an optimal policy for scheduling transmissions of associated users, whose implementation is discussed next.

D. Implementation of R-Cloud

We implemented the R-Cloud system with N200 radios and UHD USRP universal driver developed by Ettus Research [24], We chose to use the open-source GNU Radio v3.7 [10] as the software development platform. The system was built on top of Bastian Bloessl's gr-ieee802.11 PHY layer [1]. For the backhaul connectivity, we used TCP sockets to carry messages between the cloud server and BSs. We leveraged multithreaded programming for our implementation such that each user will be served by a separate thread at runtime.

We implemented the proposed cloud-level optimizer for coarse-grained control at the cloud level. We used Gurobi [13] to solve the LP instances described above. The cloud optimizer runs Algorithm 1 every 10 seconds to ensure that it is responsive to runtime fluctuations, while keeping the overhead at a reasonable level.

We also implemented the BS scheduler for fine-grained control at the BS level. We leveraged Matlab CVX [8] to solve the CP-BS instances described above. In our implementation, we used multiple queues to buffer data frames from different users and randomly selected frames to transmit with probabilities given by the scheduling policy (mentioned above) in the BS-level scheduler. A BS-level scheduler also keeps estimating the runtime average data rate ($\bar{r_b}$) by calculating a weighted moving average of instantaneous measurements and reports it to the cloud optimizer for decision making at the cloud level. Moreover, when a user association changes, the scheduler of the current BS will inform the affected user of the channel used by the new BS such that the user can tune its channel accordingly to complete the handover.

In addition, we implemented MAC framing on top of Bastian Bloessl's gr-ieee802.11 PHY layer. Specifically, at runtime, a sender encapsulates data payload with a unique MAC header (including preamble, duration, src_mac_addr, dest_mac_addr, FCS, etc) to produce a MAC frame. Correspondingly, a receiver strips off the MAC header and checks if the destination MAC address matches its own address.

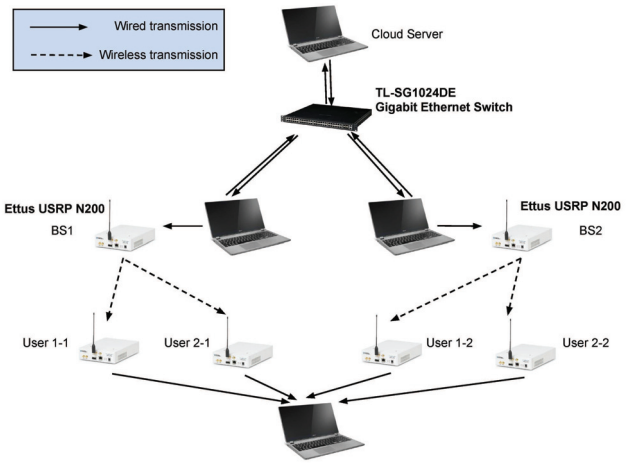


Fig. 3: The testbed for evaluating R-Cloud

IV. Performance Evaluation

We evaluated R-Cloud comprehensively via both experiments over the testbed and simulation. In this section, we discuss our testbed setup first, and then we present and analyze our experimental and simulation results.

A. Testbed Setup

We set up a testbed (shown in Fig. 3) in our lab for experiments based on the implementation described above. It consists of 4 System76 [22] laptops running ubuntu linux 12.04LTS, 6 N200s radios, each of which is equipped with a SBX Tx/Rx daughterboard and a VERT2450 dual band (2.4–2.5 and 4.9–5.9 GHz) omni-directional antenna. Communications between BSs and the cloud server were supported by a TP-Link TL-SG1024DE switch [23]. In this testbed, a laptop serves as the cloud server; two laptops are used to control operations of two radios that serve as BSs. Four radios are used to emulate four different users, which are all controlled by a single laptop. For the performance evaluation purpose, we configured the testbed with two MVNOs, each of which has 2 users. In Fig. 3, user 1-2 refers to user 2 of MVNO 1. In addition, we configured two BSs to work on two different channels on the 5GHz band respectively.

B. Experiments On the Testbed

In all experiments over the testbed, we engaged 2 MVNOs and 4 users (U1-U4), where U1 and U3 belong to MVNO1, U2 and U4 belong to MVNO2. Each user can be associated with any of 2 BSs. We developed a packet generator to generate packets for all 4 users at a rate higher than the maximum arrival rate of a user queue, i.e., traffic load is heavy enough to keep the network always busy.

1) Effective Radio-as-a-Service (RaaS): Experiment 1: In this experiment, we aimed to show that how resource reservation ratio γ_v affects the actual throughput. We compared the throughput of MVNO1 with that of MVNO2. We set $\epsilon_1 = 20\%$ and $\epsilon_2 = 15\%$. We increased γ_1 from 10% to 80% with a step of 10% while keeping γ_2 at a constant of 20%. From Fig. 4 we can see, a larger γ_1 leads to higher throughput of MVNO1 while the throughput of MVNO2 retains almost the same throughout the experiment. It shows that MVNO can obtain higher throughput by reserving more resources (higher γ_v).

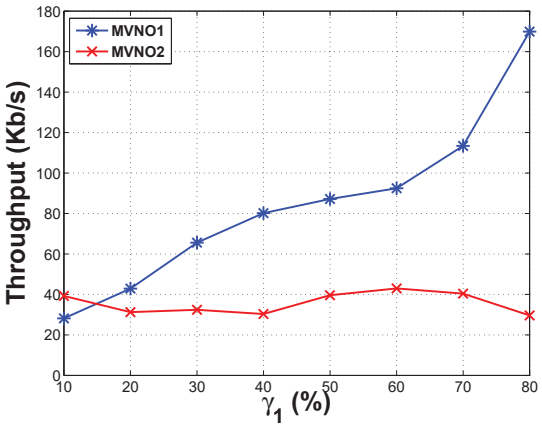
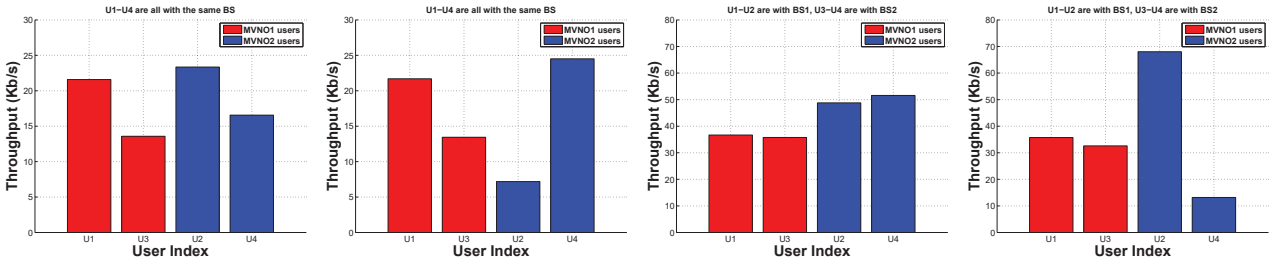


Fig. 4: Experiment 1: throughput VS. resource reservation ratio

2) Performance Isolation: Experiment 2: This experiment was designed to validate the performance isolation between MVNOs at a BS, i.e., user demands change in one MVNO will not affect throughput of other MVNOs. We assigned U1-U4 to the same BS and set $\gamma_1 = 45\%$, $\gamma_2 = 55\%$. The results are shown in Fig. 5(a) and Fig. 5(b). Initially, users had different demands (e.g. U1 - 20Kb/s, U2 - 22Kb/s, U3 - 12Kb/s, U4 - 15Kb/s). We can see from Fig. 5(a), R-Cloud ensures the satisfaction of their demands. Note that the actual throughput might be larger than that the demand, which is the minimum data rate a user requests to have. Then we dropped U2's demand to 5Kb/s and boosted U4's demand to 24Kb/s. From Fig. 5(b), we can find that the throughputs of U2 and U4 change accordingly. However, comparing Fig. 5(a) with Fig. 5(b), we observe that the throughputs of U1 and U3 do not change. This shows that R-Cloud leads to performance isolation between MVNOs at a BS.

Experiment 3: We designed this experiment to verify the performance isolation between MVNOs across the entire network (i.e. multiple BSs). We assigned U1 and U2 to BS1; and U3 and U4 to BS2. We set $\gamma_1 = 40\%$ and $\gamma_2 = 60\%$. The results are shown in Fig. 5(c) and Fig. 5(d). Initially, U1–U4 had different demands (U1 - 32Kb/s, U2 - 45Kb/s, U3 - 32Kb/s, U4 - 48Kb/s). Fig. 5(c) shows that demands of all users can be satisfied. Then, we increased U2's demand to 64Kb/s and also dropped U4's demand to 10Kb/s. We can observe from Fig. 5(d), throughputs of users of MVNO2 change accordingly. However, the throughputs of MVNO1's users (U1 and U3) do not change. This demonstrates the performance isolation between MVNOs across the entire network. Furthermore, Fig. 5 shows that R-Cloud tries its best to support the demands of all the users and MVNOs by dynamically adjusting their resource allocation according to their demands.

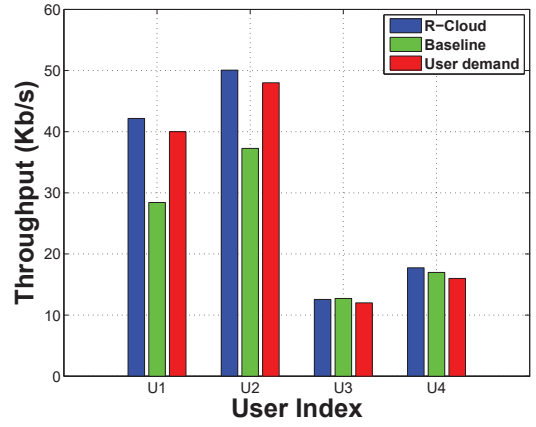


(a) (b) (c) (d)

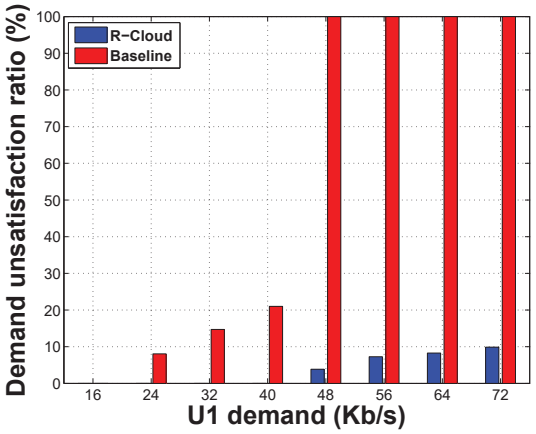
Fig. 5: Experiments 2&3: performance isolation between MVNOs. (a),(b) at a BS; and (c),(d) across multiple BSs

3) Cloud-level optimizer: Experiment 4: We designed this experiment to demonstrate the performance of the cloud-level optimizer in R-Cloud. We compared it with a baseline solution in which user association is determined in a traditional way (i.e., a user is associated with the BS offering the highest signal strength) then resource allocation ratios $< x_{b,v} >$ can be computed by solving an LP as discussed above. We placed U1 and U2 close to BS1 while U3 and U4 close to BS2 such that U1 and U2 would be associated with BS1 while U3 and U4 with BS2. We set $\gamma_1 = 40\%$, $\gamma_2 = 30\%$, $\epsilon_1 = 25\%$, $\epsilon_2 = 20\%$, $r_2 = 48$Kb/s, $r_3 = 12$Kb/s and $r_4 = 16$Kb/s. The results are shown in Fig. 6(a) and Fig. 6(b). Fig. 6(a) illustrates the throughput of each user against the corresponding demand, in which $r_1 = 40$Kb/s. It can be observed that demands of U1 and U2 cannot be satisfied by the baseline while they are satisfied by the proposed approach. This is because, in the baseline, U1 and U2 are both associated with BS1 but the aggregated demand exceeds its capacity. However, in R-Cloud, user association is optimized to meet user demands: U1 and U4 are associated with BS1 while U2 and U3 are assigned to BS2.

We increased r_1 from 16Kb/s to 72Kb/s with a step size of 8Kb/s. Fig. 6(b) shows the performance of cloud-level optimizer in terms of demand unsatisfaction ratios. When the baseline is used, increasing U1's demand (r_1) leads to BS overloading. Consequently, more user demands cannot be satisfied. In some cases (e.g. r_1 reaches 48Kb/s), the baseline cannot even find a feasible solution of resource allocation (we marked the demand unsatisfaction ratio as 100% for such cases); while the proposed approach can still find a feasible solution with reasonable amount of unsatisfied demand. In short, compared to the baseline, the proposed cloud-level optimizer can accommodate more demands and lead to much lower unsatisfied demands by optimizing user association and resource allocation.



(a) Throughput



(b) Demand unsatisfaction ratio

Fig. 6: Experiment 4: performance of cloud-level optimizer

4) BS-level scheduler: Experiment 5: To demonstrate the performance of the proposed BS-level scheduler in R-Cloud, we compared it with a baseline algorithm, NVS-VTT [16], in terms of the utility value (i.e., the sum of product of user weight $< w_u >$ and his/her actual throughput) at a BS. NVS-VTT tags each packet that arrives at the per-flow queue with a monotonically increasing virtual time. It then selects to serve the packet tagged with minimum virtual time from the heads of flow queues of the MVNO at each scheduling instant. This policy is similar to the widely-used First-In-First-Out (FIFO) policy. We set $w_1 = 0.13$, $w_2 = 0.27$, $w_3 = 0.2$ and $w_4 = 0.4$. All users U1–U4 are associated with the same BS. From Fig. 7, we can see that the proposed BS-level scheduler is superior to the baseline in terms of both individual and total utilities, which means higher throughput and better user satisfaction.

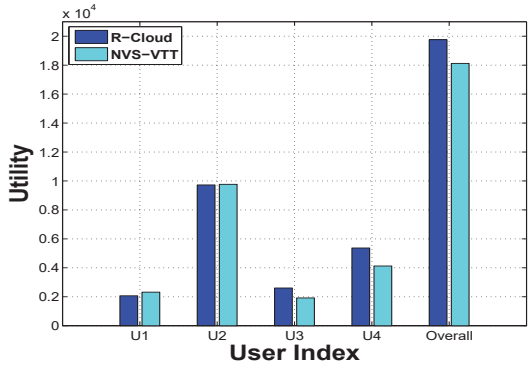


Fig. 7: Experiment 5: performance of the BS-level scheduler

TABLE II: Common Simulation Settings

Parameter	Value
Field size	$3 \times 3\ km^2$
BS runtime average data rate ($\bar{r_b}$)	80Mb/s
Max association range ($d_{\max}$)	1000m
No. of BSs	20
No. of MVNOs	3
No. of MVNO1's users	120
No. of MVNO2's users	100
No. of MVNO3's users	80
No. of user associations per iteration (k)	5
MVNO1's resource reservation ratio (γ_1)	40%
MVNO2's resource reservation ratio (γ_2)	35%
MVNO3's resource reservation ratio (γ_3)	25%
MVNO1's runtime violation ratio (ϵ_1)	25%
MVNO2's runtime violation ratio (ϵ_2)	20%
MVNO3's runtime violation ratio (ϵ_3)	15%

C. Simulation

We also evaluated the performance of R-Cloud in large cases via simulations. In our simulation, we considered a test field of 3×3 km^2 with 20 BSs uniformly deployed there. Three MVNOs were involved, where the numbers of users were set to 120, 100 and 80 respectively. All the users were randomly distributed across the field. The maximum range for user association was set to 1000m. MVNO1 reserved $\gamma_1 = 40\%$ resources across the network while MVNO2 and MVNO3 reserved $\gamma_2 = 35\%$ and $\gamma_3 = 25\%$ respectively. The runtime violation ratios were set to $\epsilon_1 = 25\%$, $\epsilon_2 = 20\%$ and $\epsilon_3 = 15\%$ respectively. We set the runtime average data rate of a BS ($\bar{r_b}$) to 80Mb/s. Table II summarizes the common simulation settings. The user demands $< r_{u,v} >$ were randomly generated in the range of $[r_{min}, r_{min} + 0.8$Mb/s$]$. We considered the scenario of varying r_{min} from 0.6Mb/s to 1.3Mb/s with a step size of 0.1Mb/s. The simulation results are shown in Figs. 8–10.

In Fig. 8, we compared R-Cloud with a baseline approach in terms of demand unsatisfaction ratio. In the baseline, user association is easily determined by always assigning a user to the closest BS and resource allocation can then be calculated by solving an LP mentioned above. We can see that R-Cloud can satisfy more user traffic demands than the baseline. In some cases (e.g. when r_{min} reaches 1.1Mb/s), no feasible solution can be found by the baseline (we marked the demand unsatisfaction ratio as 100% for such cases) while R-Cloud can still return feasible solutions with fairly low demand unsatisfaction ratios. These results justify the effectiveness and flexibility of R-Cloud for supporting RaaS.

Fig. 9 and Fig. 10 illustrate the resources allocated to MVNO 1 across the network of 20 BSs given by the baseline and R-Cloud respectively. After user association $< y_{u,v,b} >$ is determined by either the baseline or R-Cloud, each BS has an aggregate demand measured in Mb/s. In order to represent the amount of resources it needs to fully satisfy the aggregate demand on each BS, we define Resource Demand Ratio (RDR) that is calculated by the following equation:

$$RDR_b = \frac{\sum_{u \in U_v} r_{u,v} y_{u,v,b}}{\bar{r_b}}, \forall b \in B, \forall v \in V. \quad (18)$$

We compared resource allocation ratio against resource demand ratio on every single BS to demonstrate how effectively BS resources are allocated to MVNO1. It can be observed that R-Cloud leads to more balanced and reasonable resource allocation (blue bars and corresponding red points fit better in Fig. 10). Specifically, for the baseline, since each user is always associated with the nearest BS, it is likely that some BSs are overloaded while some stay under-utilized. Overloaded BS will cause unsatisfaction of some user demands while under-utilized BSs lead to resource wastage. It can be observed from Fig. 9 that some BSs (e.g. 1, 5, 10, 11, 14, 16 and 18) are overloaded while some BSs (e.g. 4, 7, 8, 9, 12, 13, 15, 17 and 19) stay under-utilized. Compared to the baseline, R-Cloud performs better. From Fig. 10, we can see that R-Cloud helps balance loads and wisely allocate resources to the users across all the BSs according to their demands such that the demand of every user can be satisfied.

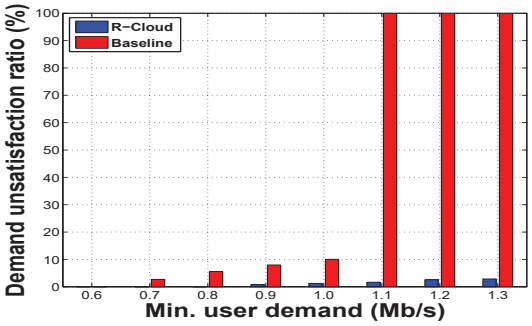


Fig. 8: Simulation: demand unsatisfaction ratio

V. Conclusions

In this paper, we presented design, implementation and evaluation of a novel cloud framework, R-Cloud, to enable RaaS. R-Cloud employs a hybrid two-level control framework to enable coarse-grained and fine-grained resource allocation at cloud and BS levels respectively. Our design of R-Cloud leads to the following desirable features: 1) hybrid control

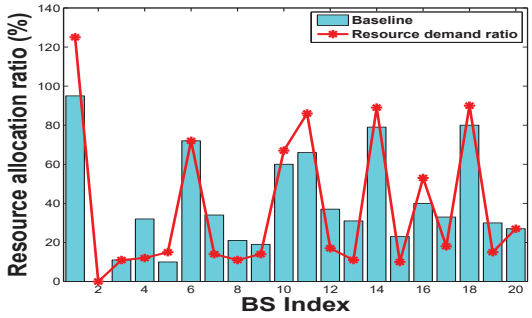


Fig. 9: Simulation: resource allocation of MVNO1 across all the BSs given by the baseline

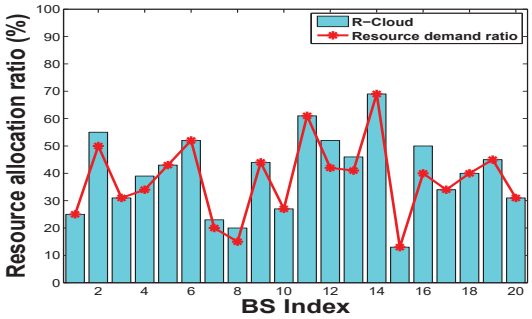


Fig. 10: Simulation: resource allocation of MVNO1 across all the BSs given by R-Cloud

for resource allocation; 2) effective resource sharing at the cloud level; 3) optimal transmission scheduling at a BS; and 4) performance isolation. Specifically, R-Cloud optimizes resource sharing and user association using an LP rounding based algorithm at the cloud level; and it effectively shares resources among multiple MVNOs at a BS using an optimal scheduling policy. We implemented R-Cloud over a wireless network testbed with software-defined radios. It has been shown by extensive experimental and simulation results that R-Cloud can achieve effective RaaS over wireless networks and the proposed resource allocation algorithms outperform widely-used baselines.

References

[1] Gr-ieee802-11, http://www.ccs-labs.org/software/ gr-ieee802-11/.

[2] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri, SplitAP: leveraging wireless network virtualization for flexible sharing of WLANs. *Proc. of IEEE Globecom*, Miami, Florida, USA, Dec. 2010.

[3] G. Bhanage, I. Seskar, R. Mahindra, and D. Raychaudhuri, Virtual basestation: architecture for an open shared WiMAX framework, *Proc. of ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, New Delhi, India, Aug. 2010.

[4] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo. CloudIQ: a framework for processing base stations in a data center. *Proc. of ACM Mobicom*, Istanbul, Turkey, Aug. 2012, pp. 125–136.

[5] S. Boyd and L. Vandenberghe. *Convex optimization*, Cambridge, UK: Cambridge University Press, 2004.

[6] C. S. Chang, Stability, queue length, and delay of deterministic and stochastic queuing networks. *IEEE Transactions on Automatic Control*, vol. 39, no. 5, May 1994, pp. 913–931.

[7] C. M. R. Institute, C-RAN: the road towards green RAN, *White Paper*, Oct. 2011.

[8] CVX research, *http://cvxr.com/cvx/*.

[9] USRP networked series, *http://www.ettus.com/ product/category/USRP-Networked-Series/*.

[10] GNU Radio, *http://gnuradio.org/redmine/projects/gnuradio/wiki/*.

[11] A. Gudipati, *et. al.*, SoftRAN: software defined radio access network. *Proc. of ACM SIGCOMM workshop on HotSDN*, Hongkong, China, Aug. 2013, pp. 25–30.

[12] K. Guo, S. Sanadhya, and T. Woo, ViFi: virtualizing WLAN using commodity hardware. *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 3, 2015, pp. 41–48.

[13] Gurobi Optimization, *http://www.gurobi.com/*.

[14] A. Helmy and T. Le-Ngoc, Low-complexity QoS-aware frequency provisioning in downlink multi-user multicarrier systems. *in Proc. of IEEE Wireless Commun. and Networking Conf. (WCNC)*, Apr. 2014, pp. 1785–1790.

[15] X. Jin, L. E. Li, L. Vanbevery, and J. Rexford, SoftCell: scalable and flexible cellular core network architecture, *Proc. of ACM CoNEXT*, 2013, pp. 163–174.

[16] R. Kokku et al. NVS: a substrate for virtualizing WiMAX networks, *Proc. of ACM MobiCom*, Chicago, Illinois, USA, Sep. 2010, pp. 233–244.

[17] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, CellSlice: cellular wireless resource slicing for active RAN sharing. *Proc. of International Conference on Communication Systems and Networks*, Bangalore, India, Jan. 2013.

[18] L. E. Li, Z. M. Mao and J. Rexford, Toward software-defined cellular networks, *Proc. of IEEE/ACM EWSDN*, 2012, pp. 7–12.

[19] R. Mahindra *et. al.*, Network-wide radio access network sharing in cellular networks. *Proc. of IEEE ICNP*, Gottingen, Germany, Oct. 2013.

[20] K. Nakauchi, Y. Shoji, and N. Nishinaga. Airtime-based resource control in wireless LANs for wireless network virtualization. *Proc. of IEEE ICUFN*, Phuket, Thauland, July. 2012, pp. 166–169.

[21] K. Sundaresan, M. Y. Arslan, S. Singh, S. Rangarajan, and S. V. Krishnamurthy, FluidNet: a flexible cloud-based radio access network for small cells, *in Proc. of ACM MobiCom*, Miami, Florida, USA, Sept. 2013, pp. 99–110.

[22] System 76 laptops, *https://system76.com/laptops/gazelle/*.

[23] 24-port gigabit easy smart switch, *http://www.tp-link.com/en/products/details/cat-41_TL-SG1024DE.html/*.

[24] USRP hardware driver software, *http://code.ettus.com/redmine/ettus/projects/uhd/wiki/*.

[25] D. Wu and R. Negi, Effective capacity: a wireless link model for support of quality of service, *IEEE Transactions on Wireless Communications*, vol.2, no. 4, July 2003, pp. 630–643.

[26] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, OpenRAN: a software-defined ran architecture via virtualization. *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, 2013, pp. 549–550.

[27] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, LTE mobile network virtualization. *Mobile Networks and Applications*, vol. 16, no. 4, 2011, pp. 424–432.

[28] L. Zhao, M. Li, Y. Zaki, A. Timm-Giel, and C. Gorg, LTE virtualization: From theoretical gain to practical solution. *Proc. of International Teletraffic Congress*, San Francisco, USA, Sep. 2011, pp. 71–78.