Network-Wide Radio Access Network Sharing in Cellular Networks

Rajesh Mahindra*, Mohammad A. (Amir) Khojastepour*, Honghai Zhang*†, Sampath Rangarajan*
*NEC Laboratories America Inc., Princeton
{rajesh, amir, sampath}@nec-labs.com
†Currently at Google
honghaiz@gmail.com

***Abstract*—Mobile operators are witnessing a dramatic increase in traffic spurred by a combination of popularity of smart-phones, innovative applications and diverse services. As mobile traffic transitions from being voice dominated to video and data dominated, the revenue per byte for the mobile operators is declining at an unhealthy rate. To counter the traffic growth and build cost-effective networks, many operators are now forging alliances for RAN (Radio Access Network) sharing to improve coverage and capacity at reasonable investments and operational costs. This paper presents the design and implementation of NetShare, a *network-wide* radio resource management framework that provides effective RAN Sharing. NetShare introduces a novel two-level scheduler split between the mobile gateway and the cellular basestations to effectively manage and allocate the wireless resources of the radio access network composed of multiple basestations among multiple different entities (such as operators, content providers, etc.) that share the network. Firstly, NetShare provides performance isolation across entities with a minimum guaranteed resource allocation to each entity across the network. Secondly, NetShare optimally distributes the resources to each entity across the network proportional to the resource demand at each basestation. Through extensive LTE-based system simulations and prototype evaluations on a WiMAX testbed, we show the efficacy of NetShare in (a) providing isolation across entities and (b) efficiently distributing resources for each entity across the network thus achieving high utilization of resources for an entity.**

I. INTRODUCTION

In line with growing costs and declining revenues for mobile operators (MNOs), network sharing is emerging as a disruptive mechanism to control network deployment costs. Although several components of the mobile network can be shared, RAN (Radio Access Network) Sharing is believed to have the most impact [1]. While passive RAN sharing involving common cell sites, power generators etc. is currently prevalent among MNOs, active RAN sharing involving sharing of wireless basestation components is steadily gaining popularity. More recently with the advent of LTE, there is growing interest among MNOs to additionally share the spectrum with other MNOs [2]. This is evident from the emergence of new business models including wholesale networks such as [3] and [4]. Additionally, spectrum sharing would increase spectral efficiency as measurement studies have shown that spectrum resources of a single operator are often under-utilized in many areas [5]. Hence, regulatory bodies are actively considering spectrum sharing among MNOs to improve statistical multiplexing and utilization of wireless resources [6]. Besides MNOs sharing networks for cost savings and faster rollouts, other *entities* such as mobile virtual network operators (MVNOs), CDNs and Service providers, who do not own the network infrastructure or spectrum, can also reserve a part of a MNOs network to provide specific services to their users. Techniques for RAN sharing can also be applied by MNOs to provide more sophisticated group and enterprise data plans with bandwidth guarantees over their networks; such plans can generate additional revenue for the MNOs. In this paper, we focus on techniques that provide effective RAN sharing utilizing *spectrum sharing*.

While commercial basestations provide sophisticated mechanisms to schedule traffic flows, existing scheduling frameworks do not provide effective radio resource management techniques for spectrum sharing among several entities. Recently, there have been a few efforts on wireless resource virtualization [7]–[9] that aim to provide effective resource management solutions for sharing basestations. Such techniques provide performance isolation across entities by enforcing a minimum guarantee of resources for each entity at each basestation. However these techniques are restrictive in terms of allocating basestation resources to the different entities across the network since: (1) The aforementioned per-basestation techniques fail to provide mechanisms for an entity to control its aggregate resource allocation across an *entire network of basestations*: the user distribution, average user-channel conditions and user-traffic requirements of an entity may vary significantly across basestations over time. Enforcing static per-basestation resource reservation may not meet the dynamic requirements of the users of an entity at some of the basestations in the network. (2) These techniques do not guarantee a minimum aggregate resource allocation across the network: Firstly, estimating and defining the average resource requirements for each entity on a per-basestation level may not be practically viable as the requirement varies over time and area [5]; defining the resource requirement over a specific geographical area that is potentially covered by several basestations would be a more realistic option. Secondly, per-basestation reservation may lead to lower resource utilization across the network for an entity.

Our Approach: NetShare aims to provide an effective RAN sharing technique that strives to manage and schedule the wireless resources at multiple basestations across different entities sharing the network. In this respect, NetShare is designed as a robust resource management algorithm that strives to achieve the following high-level goals: (1) Isolation:

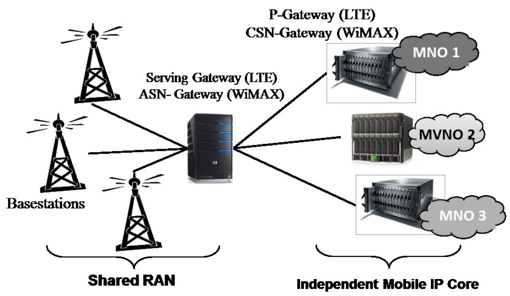


Figure 1. MOCN RAN-sharing model.

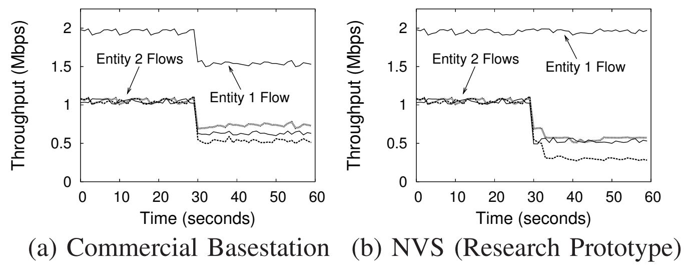


Figure 2. Current basestation schedulers.

NetShare ensures that an entity receives a chosen fraction of the wireless resources across a set of basestations in the network. (2) Programmability: NetShare exposes an interface for the entities to control the distribution of their resource allocation across the network by allowing the entities to define or compute the resource demand of their users' flows at each basestation.

Contributions: In this paper, we design and implement the NetShare system that achieves effective network-wide RAN sharing. To meet the above mentioned goals of performance isolation and effective resource distribution for each entity in the presence of RAN sharing, NetShare's design brings-forth two novel contributions:
(a) To achieve a light-weight and simple design, NetShare is designed as a split architecture with (1) a central gateway-level component that ensures network-level resource isolation and optimal resource distribution for each entity at a course time-scale and (2) distributed basestation-level scheduler that leverages a previously proposed basestation virtualization technique NVS [7] to ensure isolation across each entity at a link-level at fine time-scales.
(b) The design of NetShare is equally applicable across multiple OFDMA based fourth-generation cellular technologies including LTE, WiMAX, and LTE-Advanced.

We have evaluated NetShare using large-scale LTE simulations and a prototype on a WiMAX testbed. We compare its performance with a system that implements NVS on each of the basestations independently. As one example, our experiments show that NetShare improves an entity's resource utilization to 40% from 30% achieved with static per-basestation reservation scheme. To the best of our knowledge, this is the first detailed design, implementation and evaluation of a system that provides effective Network-wide RAN sharing.

II. Background

In this section, we present a brief background on cellular network architectures and active RAN sharing.

A. Cellular Background

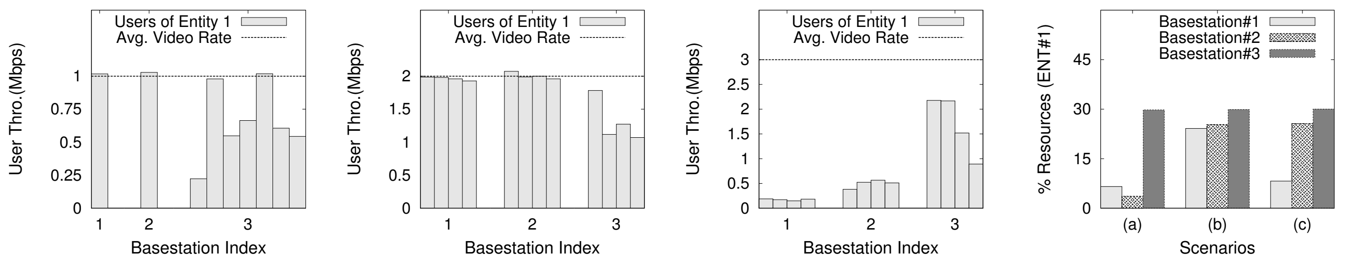
The cellular network architecture consists of two parts: the Mobile IP Core and the Radio Access Network (RAN). Using LTE as an example, the Mobile Core, including the Serving Gateway (S-GW) and Packet Data Network Gateway (PDN-GW), provides the functionalities of IP connectivity, authentication, authorization and accounting (AAA), and support for roaming. The S-GW typically handles and routes traffic to and from hundreds of basestations. WiMAX employs a similar architecture except that the S-GW and PDN-GW are replaced by ASN-Gateway and CSN-Gateway, respectively. The RAN includes basestations (or eNodeBs) that perform RRM (Radio Resource Management). Basestations incorporate downlink and uplink MAC (flow) schedulers which achieve efficient wireless resource allocation across multiple user flows. In both LTE and WiMAX, wireless radio resources are OFDMA frames (or subframes) which are divided into "resource blocks" or "slots" in the time and frequency domain. Throughout this paper, we refer to radio resources simply using the term "*resources*".

B. Network Sharing in 3GPP

In meeting the goals of network sharing sought by MNOs, the 3GPP mobile broadband standard has identified two reference architectures [10]: (1) Gateway Core Network (GWCN) configuration: Both the core network and the RAN components are shared. (2) Multi-Operator Core Network (MOCN) configuration: The core networks are operated by different entities and only the RAN components are shared, as shown in Figure 1. Under both architectures, the MNOs may (a) share the basestation hardware and software components, but use separate spectrum (orthogonal frequencies) or (b) share the spectrum (wireless resources) in addition to the basestation components. Our focus is on the latter model of RAN sharing in which the spectrum is shared among multiple entities.

III. Problem Formulation

Although, there exist several RAN sharing deployments today, for example, Telefonica O2 and Vodafone in Germany, Telenor and Hutcheson in Sweden, etc., where the operators pool their spectrum resources, current infrastructure deployments simply adhere to the 3GPP standards employing certain ad-hoc techniques to avoid extreme level of interference across entities: (a) Most MNOs employ admission control of flows from different entities to enforce isolation; (b) Certain MNOs rely on shaping flows of the MVNOs that they host to avoid affecting the quality of experience of its own users. The above mentioned schemes may work to some degree for today's voice dominated networks, but as recent trends show an increase of data traffic on cellular networks, admission control and/or traffic shaping itself will not suffice. To demonstrate this shortcoming, we conduct a simple experiment with two entities sharing a basestation on a commercial WiMAX basestation. Entity 1 has a single flow while three flows of Entity 2 are admitted to the basestation. To ensure that Entity 1 meets it requirement of 2 Mbps irrespective of changes to traffic patterns of users of Entity 2, the flows of Entity 2 are shaped at 1 Mbps each. This works for a particular configuration as



(a) Distinct # users (b) Distinct channel cond. (c) Distinct traffic req. (d) Res. utilization of Entity 1

Figure 3. Case for NetShare.

shown in Figure 2(a) for the initial 30 seconds. However when the users of Entity 2 are moved away from the basestation starting at 30 seconds, the signal strength of their links degrade. Hence as shown in Figure 2(a), the throughput of the users of Entity 2 drops to around 600 Kbps. More importantly, the throughput of the user of Entity 1 also drops despite no change in its configuration. This effect is because current basestation schedulers are designed to achieve flow-based resource fairness and hence fail to provide effective resource management to enable RAN sharing.

A key component of active RAN sharing is scheduling wireless resource allocation across the basestations such that each entity receives at least its reserved share. Such sharing may be realized by reserving resources for entities at every basestation by using per-basestation techniques as in [7]–[9]. For instance, if an entity desires to reserve 30% of the network, every basestation in the network is statically configured to reserve 30% of its resources for this entity. As seen in Figure 2(b), one such prior work (NVS) [7] provides isolation across the two entities for both configurations. However, such techniques based on per-basestation resource provisioning for entities have certain shortcomings:

(a) Such designs are *inflexible* in terms of dynamically redistributing the resources of an entity effectively across the network to meet the requirements of its individual users spread across the network. We further motivate this point with experiments performed over a WiMAX prototype set up with a network of three basestations shared by two entities. NVS is implemented on every basestation such that the two entities 1 and 2 have 30% and 70% resource reservation respectively. Entity 2 has a fixed set of 12 users uniformly distributed across the basestations. The experiment is repeated for three different configurations for the users of Entity 1. In each case, we plot the per-user throughput for all users of Entity 1. As seen in Figures 3(a),(b) and (c), the users of Entity 1 at basestation 3 never meet their requirements although Entity 1 is lightly loaded at basestations 1 and 2. Specifically, in Figure 3(a), Entity 1 has several users in basestation 3 while it has only a single user at basestations 1 and 2. The resource allocation of 30% at basestation 3 is insufficient for the users of Entity 1 to meet their requirement of 1 Mbps average rate. In the second case in Figure 3(b), since the users at basestation 3 are placed far from the basestation, they have poor channel conditions and need more resources to meet their bandwidth requirement of 2 Mbps. Although, users of Entity 1 at basestations 1 and 2 use less than their reserved 30% of the resources, Entity 1 does not have the flexibility to redistribute those resources to basestation 3. Similarly in the final case in Figure 3(c), users at basestations 1, 2 and 3 receive videos with average rates of 100 Kbps, 500 Kbps, and 3 Mbps respectively. As seen from Figure 3(c), users of Entity 1 at basestation 3 do not get sufficient throughput to support their video rates and hence would receive poor video quality. Since the traffic of Entity 1 is much higher in basestation 3 than in basestations 1 and 2, the requirement of wireless resource exceeds its reservation of 30% in basestation 3.

(b) The aggregate *resource utilization* of the entity across the network suffers with static per-basestation resource reservation schemes. To show this effect, we plot the resource utilization of Entity 1 at all the basestations for the three cases explained earlier. Figure 3(d) shows that Entity 1 does not utilize its share of reserved resources (i.e. 30%) at basestations 1 and 2 in all the three cases despite having sufficient demand at basestation 3. Hence, the failure of such schemes to guarantee aggregate resource allocation across the network can cause loss in resource utilization for an entity across the network. As a result, the entity does not meet its reserved resource fraction of 30% across the network.

A. Objectives

In meeting the goal of scheduling the aggregate resources of an entity across a set of basestations, NetShare is designed based on three key objectives:

1. Performance Isolation: Any change in one entity such as change in the number of users, their traffic or link conditions and mobility should not affect the aggregate resource allocation of other entities.

2. Effective Resource Distribution: NetShare tries to schedule the resource allocation of an entity across the network of basestations such that the traffic demand of that entity is met at every basestation.

3. Network Utilization: Wireless spectrum is a scarce resource and an effective RAN sharing scheme should be designed such that the resources are not under-utilized. Hence, NetShare is designed as a work-conservative scheduler such that the unused resources of an entity are allocated to other entities.

B. Service-level Agreements

NetShare is designed under the assumption that a *Network Owner* (eg, an MNO) operates the physical network and hosts several entities that share the network. To ensure proper understanding and accountability between the network owner and the entities, concrete SLAs are a prerequisite to any RAN sharing technique. Although, SLAs could include complicated pricing schemes, we base the design of NetShare on an SLA that we believe is practical and would be acceptable to the

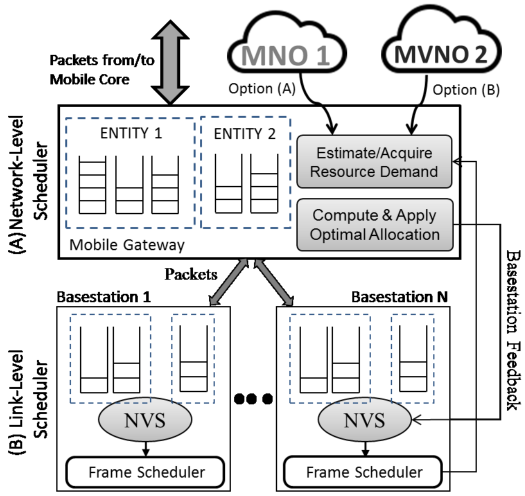


Figure 4. NetShare's software Architecture.

network owner as well as the hosted entities. In order to provide performance isolation, NetShare enables entities to make reservations for the aggregate resources on a network of basestations. Consider a network of B basestations with total resources R across the network. Let the minimum aggregate resources reserved by an entity j be L_j. For instance, if an entity is interested in reserving 30% of the aggregate resources of the network, then NetShare sets L_j to be $(0.3)\times$R. To meet all entities' reservation without violating the total resource constraint, NetShare performs admission control of entities to ensure that $\sum_j L_j = \gamma \times R$. Where γ is the overbooking factor such the $\gamma \geq 1$ to exploit statistical multiplexing. In case the network is partially reserved i.e. $\sum_j L_j < R$, for allocation purposes NetShare proportionally increases the reservation of each entity such that $\sum_j L_j = R$.

In addition to a minimum guarantee across the network, the SLA also includes a provision for an entity to define a minimum guarantee per basestation l_j^b, where b is the basestation index. This ensures that the entities have the option to receive a minimum allocation at every basestation avoiding the risk of starving flows at certain basestations. The values of l_j^b are set such as to ensure $\sum_b l_j^b < L_j$. Despite the benefits of resource distribution, certain entities may desire static per-basestation reservation. Such entities can set the values of l_j^b such that $\sum_b l_j^b = L_j$ to avoid distribution of resources across the network.

IV. Design Overview

NetShare is designed as a split hierarchical scheduler, with one component implemented within a mobile core gateway (e.g., S-GW in LTE or an ASN gateway in WiMAX) and another component present in each basestation, as shown in Figure 4. The former component is termed as the *network-level* scheduler that computes and applies the resource allocation for each entity at every basestation at course time-scales. The latter component is a *link-level* scheduler that is part of the flow scheduling framework at each basestation. An alternate design would be to distribute the entire logic within basestations. However, designing NetShare as a two-level scheduler with a component in a gateway has significant advantages: (1) Since traffic to all basestations in both downlink and uplink directions flows through the gateway, it is easier to estimate the resource demand of the entities at the gateway. (2) A gateway typically manages 100's of basestations, hence the solution is more centralized leading to lower messaging overhead. Alternatively, implementing the network-level scheduler in a distributed way on each basestation would require messaging among all basestations to obtain global information.

Network-level scheduler: This component periodically (i.e, at every epoch of τ units of time) performs three operations as summarized in Algorithm 1. To provide effective resource distribution for each entity, a metric d_j^b is defined for each entity j at basestation b. d_j^b represents an estimate of the average resource demand or requirement of all the existing flows of an entity at a particular basestation. In order to correctly estimate the resource demand for flows of an entity at a basestation (Step 2), the following information is required: (1) the entity to which a user flow belongs; this information is typically available in the mobile core (e.g., at the Mobility Manager Entity(MME) in LTE) and is utilized to account for the arrival rate of traffic of an entity, and (2) periodic feedback of average per-user transmission rate or MCS (modulation and coding scheme) which indicates the effective rate of a flow (e.g., bits per resource block in LTE); this parameter is typically maintained by commercial basestations. The above information is utilized to convert the arrival rate in bytes or bits per second to percentage of wireless resource demand of an entity. In the case where an entity estimates the resource demand, an interface is provided to the entity to retrieve the above information. After estimating or acquiring the resource demands for all entities, the scheduler optimally computes the resource allocation of each entity at every basestation, while making sure the aggregate allocation meets the minimum guarantee (Step 3). The details of the resource allocation are present later in section IV-A.

Link-level scheduler: To enforce resource allocation for entities at a basestation computed by Step 3, NetShare uses NVS [7] as the link-level scheduler at each basestation. NVS is a native basestation virtualization technique that provides an interface for entities to reserve wireless resources at a basestation. Specifically, NVS defines a parameter *minimum-reserved-resource* for every entity that desires to reserve a fraction of the total basestation wireless resources. NVS schedules the packets such that each entity receives at least its minimum reservation. NVS also ensures efficient utilization of wireless resources by allocating unused resources of an entity to other active entities in a basestation. NVS achieves this isolation by introducing the notion of *slice*: a slice is a group of flows that belong to an entity and a slice scheduling algorithm is employed to schedule flows across different slices. Since NVS is a basestation

Algorithm 1 NetShare (Network-level scheduler)

1: **Repeat** every τ units of time
2: Estimate/Aquire the average resource demand of every entity in each basestation.
3: Compute the optimal resource allocation of every entity in each basestation while meeting the aggregate resource reservation of every entity.
4: Enforce the resource allocation computed in previous step using NVS (Link-level scheduler) at every basestation.

solution, it operates at millisecond granularity (typical 10 ms for LTE and 5 ms for WiMAX) and can handle wireless link capacity fluctuations at fine time scales. Moreover, it provides effective uplink scheduling. To enable enforcement of resource allocation for entities at a basestation, the Network-level scheduler component of NetShare dynamically configures the *minimum-reserved-resource* parameter for each entity at the NVS instances running in the basestations. Note that by setting the *minimum-reserved-resource* parameter of NVS, NetShare retains the work-conservative property of NVS. For instance, if NetShare allocates 30% of the resources at a particular basestation to an entity which does not have sufficient traffic in the next epoch to meet that reservation, NVS scheduler would re-distribute the unused resources of the entity to other entities on that basestation.

A. Resource Allocation

As part of the Network-level scheduler, the resource allocation algorithm (Step 3 in Algorithm 1) forms the core component of NetShare as it ensures: (a) a guaranteed minimum resource allocation L_j to each entity across the network, and (b) optimally distributes the resources of each entity across the network as captured by the objective function of Problem 1 below which is subsequently explained in more detail. Given a set of J entities sharing a network with B basestations, let variables t_j^b represent the resource allocation to be computed by NetShare for entity j on basestation b. The resource allocation problem is formalized as follows.

Problem 1:
$$\max_{t_j^b} \sum_{b=1}^{B}\sum_{j=1}^{J} G_{j,b}(t_j^b) \tag{1}$$
$$\text{s.t. } L_j \le \sum_{b=1}^{B} t_j^b \le U_j \text{ for all } j \tag{2}$$
$$\sum_{j=1}^{J} t_j^b \le f_r(b) \text{ for all } b \tag{3}$$
$$l_j^b \le t_j^b \le u_j^b, \text{ for all } b, j \tag{4}$$

In the following we describe the formulation in more detail:

(a) Resource Distribution: Effective *resource distribution* implies the need to allocate the aggregate resources to an entity across the network such that the resource requirements of the users of that entity are satisfied at every basestation. To achieve this goal, the algorithm should ensure that the resources allocated to an entity across the network should be proportional to its resource demand. Achieving such allocation is challenging across all entities because (a) the demand across the network is different for different entities and (b) the resources of each basestation are physically limited. Hence, NetShare takes the following approach: it aims to allocate the resources of a basestation to the different entities proportional to their resource demand at that basestation. NetShare achieves this proportionality optimally by maximizing the following utility function for all entities across the network of basestations. Specifically, NetShare defines the utility function of an entity in the above optimization problem as:

$$G_{j,b}(t_j^b) = d_j^b \times log(t_j^b) \tag{5}$$

The above utility function is drawn from the principles of proportional fairness as employed by several previous works like [11], [12]. Hence, the utility function is defined as a log function of the allocation received by an entity (i.e. t_j^b) scaled by its resource demand d_j^b. Such a definition implies that the marginal utility of an entity at a basestation decreases as its resource allocation increases (log function), while also ensuring that its utility is directly proportional to the resource demand at that basestation (linear factor). Although a log utility function is more suited for elastic traffic and an entity may have non-elastic traffic such as video flows, studies have shown [13] that if flow admission control and dynamic adaptation at the application level are employed, the aggregate utility of an entity can be modeled as a log function (or any concave function). There may be alternative choices for the utility function $G_{j,b}(t_j^b)$ to realize resource distribution; the above formulation is simple yet effective that works well to meet the requirements.

(b) Strict Isolation: The first constraint directly follows from the defined SLA. NetShare always tries to allocate L_j fraction of resources to an entity j despite its aggregate resource demand falling below L_j. This ensures that NetShare will provide effective isolation even for arbitrary execution times (τ), since the resource demand for an entity might increase during the next epoch. NetShare restricts an entities' aggregate resource allocation to at most U_j.

(c) Network Heterogeneity: Heterogeneity in the radio access network is an important consideration while designing NetShare for cellular networks. Heterogeneity in cellular networks may manifest in different forms, for instance heterogeneity in: (a) Coverage (small vs pico vs macro cells); (b) Service area (downtown vs highway) and (c) QoE (cellular vs WiFi) etc. However, designing a solution to account for heterogeneity is hard since (a) the networks of different owners may have diverse heterogeneity, (b) different owners may perceive network heterogeneity differently; for example, one owner might consider the resources of small cells less valuable than its macro cells since the coverage of macro cells is greater while another owner may consider the resources of its small cells more valuable since they offer greater capacity. To make the design general in order to accommodate these diverse policies, NetShare introduces the notion of relative importance of the resources of different basestations in the network. Specifically, NetShare allows owners to define priority for the resources of all basestations. It then normalizes the resources of the basestations with the highest weight to 1 and all other basestations to a fraction depending on their relative weights. The function $f_r(b)$ in the above problem formulation represents the normalized resources available at basestation b. Hence, the total available resources at a basestation b is $f_r(b) \le 1$ such that total network resources $\sum_b f_r(b) = R$. This ensures that the resources given to an entity at a particular basestation are accounted proportional to the priority (defined by different weights) of the resources of that basestation.

(d) Per-BS reservation: The last constraint ensures that each entity receives at least l_j^b resources at basestation b. Although l_j^b values are set based on the SLA between the network owner and the entity, the choice of the values of l_j^b exposes an important trade-off for the entity: A higher value of l_j^b at a basestation ensures that the allocation can absorb a sudden surge in traffic

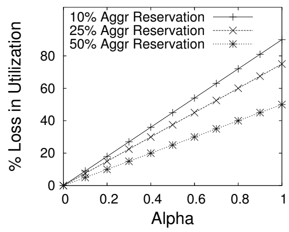


Figure 5. Tradeoff with l_j^b.

during the next epoch; however a higher value of l_j^b reduces the opportunity for NetShare to effectively distribute the resources of an entity across the network, hence affecting its resource utilization. To motivate this point, we conduct analysis for a particular formulation of l_j^b. The following formulation is also used by NetShare as the default value of l_j^b. Specifically, NetShare sets the default l_j^b to be uniform across the network as a fraction of the aggregate resource reservation L_j for that entity at every basestation:

$$l_j^b = \frac{\alpha L_j}{R} \tag{6}$$

such that $0 < \alpha < 1$. To understand the effect of higher values of l_j^b, we plot the worst-case *loss in resource utilization* P_u for an entity for different values of α in Figure 5. We defer the proof of the equation for the loss in resource utilization P_u to the appendix and instead focus on the implications. Clearly, as α increases, P_u increases irrespective of the reservation L_j. This is because as α increases, it reduces the opportunity for NetShare to redistribute the resources of the entity since it is forced to allocate a minimum resource allocation l_j^b at basestations that might have either no or very little resource demand. Note that the case where $\alpha = 1$ represents the case of static per-basestation reservation schemes like NVS. In this particular case (with L_j=25%), NetShare decreases the loss in resource utilization of an entity from about 80% with NVS to about 10% (For $\alpha = 0.1$). Although the above analysis is based on a specific formulation of l_j^b (Eq 6), the tradeoff is valid for any general formulation of l_j^b. An entity has to consider the tradeoff for minimum resource allocation at a basestation against effective network-wide resource distribution while setting its values of l_j^b. Finally, the maximum allocation of an entity at a particular basestation is limited to u_j^b.

B. Flexible Allocation Framework

To ensure that each entity can choose its own resource distribution policy, NetShare allows entities to define or compute the metric d_j^b at each basestation using two options as shown in Figure 4:

Option (A): An entity can compute the resource demand of its users at every basestation. NetShare provides an interface for such entities to provide the resource demand values to the NetShare network-level scheduler. Giving entities the flexibility to compute the resource demand is useful since certain information such as user priorities, traffic patterns and flow scheduling/optimization policies may only be known to the entity. Hence, it may be in the best interest for the entities to determine its utility of resources at different basestations. For instance, an entity may want to allocate more resources at a basestation with more *premium* users (users that have paid more for its service) or an entity may want to give more resources to a basestation with users streaming Netflix videos rather than with users streaming Youtube videos since it has the capability to transcode unencrypted video flows from Youtube but not Netflix. This option is most suited for entities such as MNOs or MVNOs that already have presence in the network.

Option (B): An entity can let NetShare compute its resource demand at each basestation. Entities such as service providers and content providers that may lack expertise in the wireless domain could choose such an option.

In initial RAN sharing deployments, we believe that Option (B) above would be widely used by entities for faster deployments and Option (A) will appear as RAN sharing evolves over time. Hence, in this section we provide guidelines for the network owners to compute the parameter d_j^b, since this parameter affects the distribution of resource allocation of an entity across the network. We set the values of these parameters in our simulation and prototype based on the following formulations.

Setting d_j^b: NetShare defines a default formulation for the resource demand d_j^b of an entity at a basestation for both non real-time and real-time traffic classes computed as:

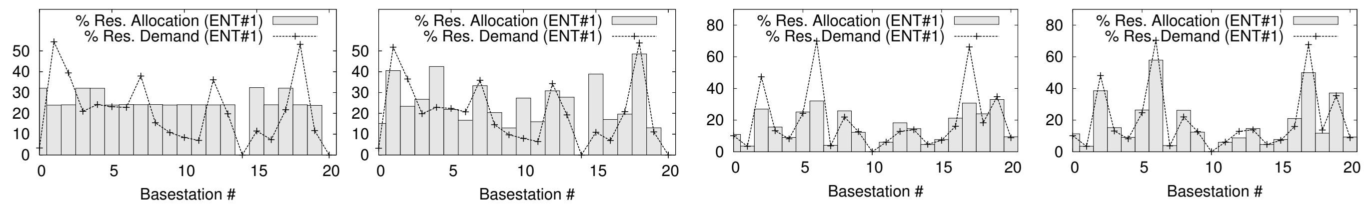
Non real-time (non-GBR):

$$d_j^b = \sum_{i \in Q_j^b} \frac{\min(\beta A_i, S_i)}{R_i} \tag{7}$$

Real-time (GBR):

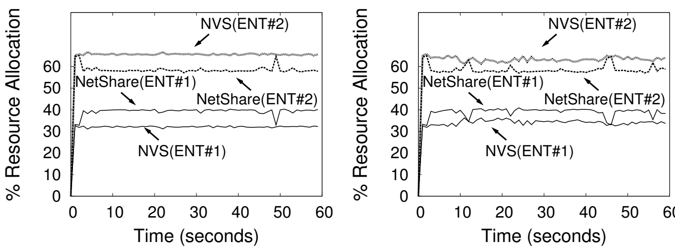
$$d_j^b = \sum_{i \in Q_j^b} \frac{M_i}{R_i} \tag{8}$$

where Q_j^b is the set of flows that belong to entity j at basestation b. In the case of real-time flows, NetShare sets the demand based on the minimum reserved rate (M_i) of that flow. In cellular networks, typically real-time flows are configured with a minimum reserved rate for flow scheduling purposes since such flows have stringent bandwidth requirements. This parameter is configured by the network based on direct or indirect knowledge from the application layer (for instance, deep packet inspection or DPI middleboxes are employed for this purpose.) On the other hand, for non real-time flows, NetShare sets the demand based on the minimum of the average arrival rate (A_i) multiplied by a factor β and the maximum sustained rate S_i, i.e. if the arrival rate of a flow is higher than its max sustained rate, then NetShare sets the demand based on the latter. β is set to > 1 so that NetShare is able to react to increase in resource demand of flows. Typically best effort flows in cellular basestations are configured with a maximum sustained rate parameter to limit the amount of resources allocated to it. To translate the bandwidth demand (say in Kbps) of flows to actual radio resource requirement, we scale the bandwidth demand by R_i, which represents the average transmission rate of the user to which the flow belongs. The transmission rate (determined by the modulation and coding scheme or MCS) of users is adapted by the basestation based on the users' average Signal-to-Noise ratio (SNR) to ensure that the user receives packets at the lowest possible loss rate. Finally, the average value of the resource demand for an



(a) NVS(FTP) (b) NetShare(FTP) (c) NVS(Video) (d) NetShare(Video)

Figure 6. Resource Distribution with NetShare.



(a) Static (b) Mobile

Figure 7. Resource Isolation with NetShare.

entity at a particular basestation is computed as the summation of the demand of all the entities' flows at that basestation.

V. Evaluation

In this section, we evaluate NetShare extensively using simulations and through a prototype implementation. In both cases, the code is written primarily in C/C++ and is around 1000 lines of code for the entire framework. To solve the resource allocation problem (Section IV-A) the Network-level scheduler in NetShare makes use of the non-linear optimization library OPT++ [14]. NetShare implementation includes a main C++ routine that sets up the resource allocation problem and the algorithm (Problem 1). It also includes several subroutines to initialize the variables, declare the constraints and define the analytical gradient and Hessian-matrix functions of the objective since the objective function in NetShare is non-linear. This routine is integrated with our simulator and prototype and it executes every τ to determine the demands for each entity and then make a call to the *interior point method* function in OPT++ library to compute the individual allocations of each entity at every basestation. In both simulations and experiments, we set τ=10 seconds, α=0.1 and β=1.1. We chose an epoch time of 10 seconds for NetShare to ensure that it is reactive to fluctuations in resource demand and capacity, while keeping the network communication overhead at a reasonable level.

A. Simulation Study

We first study the efficacy of NetShare using large-scale simulations. The simulations are performed with an in-house LTE system-level simulator in which we implemented the scheduling components. The simulator models fast fading, mobility, user arrival and several types of traffic including FTP, video and VoIP. NetShare functionality is invoked every 10 secs, and the resource allocation for the different entities is applied at all the basestations. NVS is implemented within the MAC/flow scheduler of each basestation and invoked every 10 millisecond (typical interval for LTE frame scheduling).

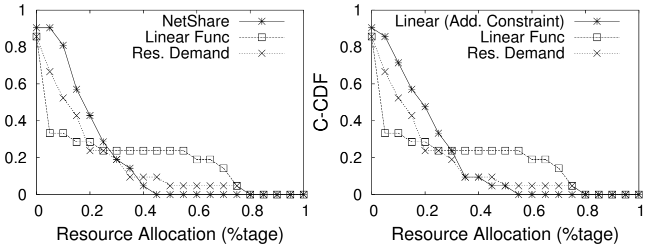
Resource Isolation: We start by demonstrating the efficacy of NetShare in providing isolation of aggregate resource allocation across entities. We set up a network of 21 basestations with 2 entities. Entity 1 reserves 40% aggregate wireless resources while Entity 2 reserves 60% aggregate resources. Entity 1 has 100 users in the network while 120 users belong to Entity 2. All users of Entity 1 stream videos of average rate 1Mbps, while users of Entity 2 download large FTP files emulating backlogged traffic. We repeat the simulations for two cases: (1) All users are stationary; (2) All users are mobile. We plot the aggregate resources allocated to both entities for these two cases in Figure 7. With NVS, every basestation tries to allocate resources in the ratio 40%:60% to Entities 1 and 2. NVS is unable to meet the aggregate resource reservation of Entity 1 since it does not have sufficient traffic on some basestations to meet its reservation. On the other hand, NetShare ensures that both entities receive their aggregate resource reservation by distributing the allocation across the network. NetShare is effective even in the case when users are mobile as seen in Figure 7(b).

Resource Distribution: In this simulation, we set up a network of 21 basestations with 4 entities. Each entity reserves 25% aggregate wireless resources. Each entity has 50 static users distributed randomly across the 21 basestations with varying channel conditions to model fast fading. We repeat the simulation for 2 cases: (1) All users download a large file using FTP to generate backlogged traffic. (2) All users stream a video of 1 Mbps average rate. Figure 6 compares the resource allocation at every basestation for one of the four entities with and without NetShare. We also plot the demand for the entities at all the 21 basestations. Demand is computed based on equation 7 for the FTP flows and equation 8 for the video flows. The plots (b) and (d) indicate that NetShare is effective in allocating resources to the entities proportional to their demand across the basestations. However, NVS allocates 25% or fewer resources to the entities at every basestation irrespective of the demand of an entity. In some basestations, note that NVS allocates more resources to the entity than its reservation of 25% because in these cases other entities in the basestations are not using their resources completely.

To further demonstrate the efficacy of the utility function employed by NetShare, we repeat the above experiment for a linear utility function and compare it with the log utility employed by NetShare. To evaluate NetShare with a linear utility function, we alter the utility function defination in Equation 5 as:

$$G_{j,b}(t_j^b) = d_j^b \times t_j^b \tag{9}$$

The constraints used are the same as in Problem 1. We plot the complementary CDF of the average resource allocation



(a) Comp. Utility Functions (b) Modifying formulation

Figure 8. Behavior of Resource Distribution.

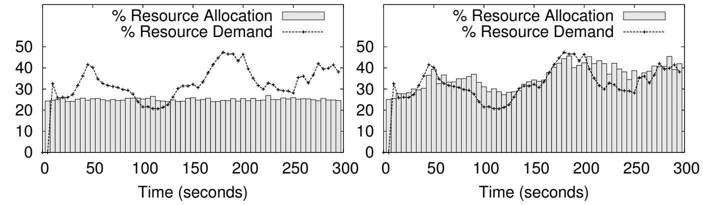
received by one of the four entities at the 21 basestations in Figure 8(a). We also plot the complementary CDF of the average resource demand of the entities across the network in Figure 8(a). As can be seen clearly from the figure, the average resource allocation with NetShare employing a log utility function is proportional to the average resource demand across the network. However, in case where NetShare employs a linear objective function, the distribution of resource allocation is biased towards basestations with high demand. This results in the entity receiving lower allocation at basestations that have low demand. For instance in this case, Entity 1 receives less than 10% allocation in almost 60% of the basestations in the network. However with the log utility function, Entity 1 receives more than 20% allocation in 60% of the basestations. Although in our simulation and implementation NetShare distributes resources proportionally to the resource demand by employing a log utility function, the framework of NetShare is general to employ any kind of utility function that meets the requirements of the network owner.

Interestingly, one of our initial designs of NetShare achieved the distribution of resource allocation of an entity proportional to its resource demand by imposing a stricter constraint. Specifically to achieve resource distribution, we modify the third constraint Equation (4) in Problem 1 to upper limit the allocation of an entity at a particular basestation (t_j^b) to its resource demand (d_j^b) at that basestation:

$$l_j^b \le t_j^b \le d_j^b, \text{ for all } b, j \tag{10}$$

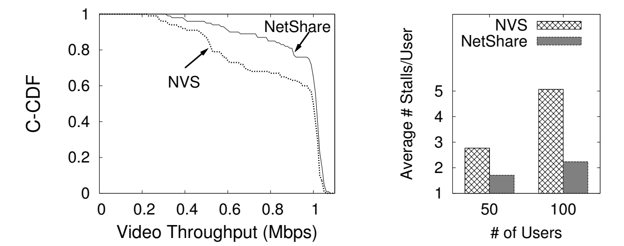
However, we found that this change makes the problem highly over-constrained and the solution is infeasible in most instances. If the problem is infeasible, the original interior point method returns a solution with minimum violation on all constraints. However, violation of the first two constraints in Problem 1 cannot be allowed in a real system. For example, if the solution violates the constraint $t_j^b \le f_r(b)$, it cannot be applied to the system directly because the total physical resource at a basestation is limited by $f_r(b)$. To address the above issue, we modified the interior point method such that it chooses to minimally violate only the constraint: $t_j^b \le d_j^b$ defined above when the solution is infeasible. Figure 8(b) shows improvement for the linear objective based resource allocation with the modified constraint. In fact, the allocation of the resoures of Entity 1 is slightly more proportional to its demand across the network than NetShare in Figure 8(a). However, we believe that this solution is unnecessarily complex and compute intensive hindering its deployment in real systems. Instead, NetShare relaxes the constraint definition to:

$$l_j^b \le t_j^b \le u_j^b, \text{ for all } b, j \tag{11}$$



(a) NVS (b) NetShare

Figure 9. NetShare with User mobility.



(a) Throughput (b) Stalls per minute

Figure 10. QoE of users with NetShare.

where u_j^b can be sufficiently large and employs a log utility in the objective (Problem 1). This leads to a compromise: a slight loss in proportional resource distribution for greater simplicity leading to a light weight solution.

User Mobility: In this simulation, we use the same setup as the previous experiment with FTP traffic. However, in this case some of the users are mobile, either at walking speeds or vehicular speeds. User mobility introduces dynamic capacity fluctuations at every basestation. We measure the performance of NetShare in the presence of such user mobility. We plot the demand and resource allocation of a single entity at a particular basestation over time in Figure 9 with and without NetShare. Clearly, NetShare is reactive in allocating resources to the entity in proportion to its demand even in the presence of high mobility whereas NVS allocates 25% resources over the entire period without reacting to changing demand. We observed similar results for the rest of the entities.

User QoE: In this experiment, we show the efficacy of NetShare in meeting the requirements of individual flows within an entity. We set up a network of 21 basestations and two entities. The first entity has 30% reservation while the second entity reserves 70% of the total resources. Entity 1 has 100 users that are randomly distributed across the basestations. Entity 2 has 63 users that are uniformly distributed across the basestations (i.e. same number of users per basestation - in this case, three users per basestation). All the users of Entity 1 stream videos of average rate of about 1 Mbps, while users of Entity 2 download FTP files. In Figure 10(a), we plot the complementary CDF of the average throughput obtained by each user of Entity 1. Clearly, users get more average throughput with NetShare than with NVS. Specifically, around 80% of the users with NetShare get almost 1 Mbps throughput required to see a stall-free video. Since NetShare allocates resources proportional to the demand, it is more effective than NVS in meeting per-flow requirements. In basestations that are congested due to higher number of users belonging to Entity 1, NVS is unable to meet their throughput demand with 30% allocation. NetShare however allocates more resources

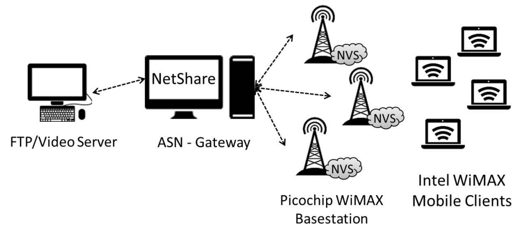


Figure 11. NetShare WiMAX prototype.

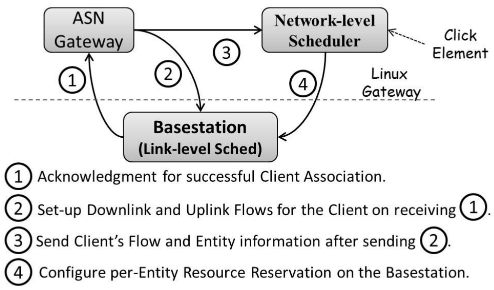
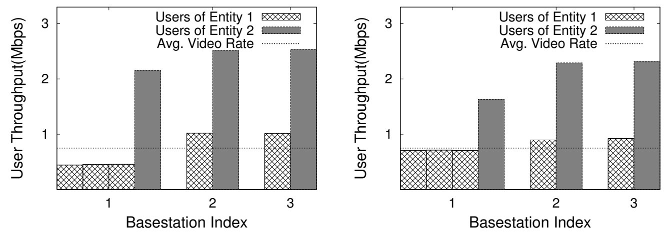


Figure 12. NetShare implementation details.

to such overloaded basestations while ensuring 30% aggregate resource allocation across the network. Additionally, to show the effect of lower throughput on QoE for video flows, we plot the average number of stalls per user per minute with and without NetShare in Figure 10(b). The results are plotted for different users in Entity 1. We employed a buffer size of 100 KB which is typically used in popular flash players today. Higher average user throughput with NetShare ensures fewer stalls while viewing the video.

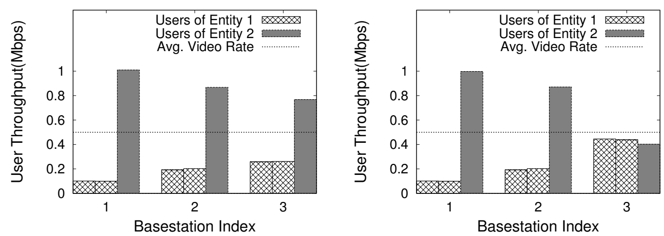
B. Prototype Evaluation

We have developed a prototype system to validate the proposed solution on a WiMAX testbed. The testbed consists of an Access Service Network (ASN) gateway, three Picochip [15] WiMAX femtocell basestations (IEEE 802.16e compliant), and several Intel WiMAX [16] clients (See Figure 11). The Network-level scheduler of NetShare is implemented as part of the ASN-gateway on a standard dual-core Linux machine directly connected to the basestations. The Link-level scheduler NVS is implemented as part of the MAC software within the basestations. As shown in Figure 12, we implement the necessary interfaces between the ASN gateway and NetShare. The ASN gateway provides an interface to the basestation for setting up service flows in the downlink and the uplink direction for each client when it registers. The service flow information including the entity it belongs to, is then passed to the Network-level scheduler that utilizes it to configure flow-to-entity mapping to perform resource allocation. We implement the Network-level scheduler functionality of NetShare as a user-level Click module [17] that intercepts all data packets from the basestation in the downlink and uplink. This lets the Network-level scheduler estimate the arrival rate of each flow and aggregate it to find the arrival rate of each entity. The Picochip basestation provides feedback on the average MCS per client to the Network-level scheduler every τ units of time. When this feedback is received, the resource allocation t_j^b for each entity j at basestation b is computed and a message containing these values is sent to all the basestations. We implement these message exchanges between the Network-level scheduler component of NetShare and the NVS instances deployed on the basestations.



(a) NVS (b) NetShare

Figure 13. Efficacy of NetShare.



(a) NVS (b) NetShare

Figure 14. NetShare with Uplink Traffic.

Downlink Experiment: We set up the three basestations being shared by two entities such that Entity 1 has 40% reservation while Entity 2 has 60% reservation. Entity 1 has 5 users who are distributed such that basestations 2 and 3 have a single user while basestation 1 has three users. Each user of Entity 1 streams a video at an average rate of around 750 Kbps. Entity 2 has a single user in all three basestations; each user downloads a FTP file of large size. As shown in Figure 13(a), NVS allocates exactly 40% of the resources to Entity 1 at basestation 1. This allocation is insufficient for users in basestation 1 to meet their requirement of 750 Kbps. However, NetShare allocates unused resources of Entity 1 from basestations 2 and 3 to Entity 1 on basestation 1. Figure 13(b) clearly shows the efficacy of NetShare in meeting the requirement of users of Entity 1 at basestation 1. Moreover, NVS allocates 31.9% aggregate resources while NetShare allocates about 38.9% aggregate resources (which is close to the 40% reservation). Thus, NetShare achieves better isolation in meeting the aggregate resource requirement of Entity 1 across the network.

Uplink Experiment: We set up three basestations to be shared by two entities such that Entity 1 has 40% reservation while Entity 2 has 60% reservation. All the three basestations transmit on orthogonal 10 MHz channels in the 2.5-2.6 GHz band. Entity 1 has two users in each of the three basestations. Users of Entity 1 at basestations 1, 2 and 3 upload video streams at rates 100 Kbps, 200 Kbps and 500 Kbps respectively. Entity 2 has a single user in all three basestations; each user uploads a FTP file of large size. NVS allocates exactly 40% of the resources to Entity 1 at every basestation. As shown in Figure 14(a), this allocation is insufficient for users in basestation 3 to meet their requirement of uploading videos of 500 Kbps. However, NetShare allocates additional resources to Entity 1 at basestation 3 since it does not use its allocated 40% at basestations 1 and 2. Again, NVS allocates only 30.77%

aggregate resources while NetShare allocates about 39.3% aggregate resources to Entity 1. Thus, NetShare achieves better isolation in meeting the aggregate resource requirement of Entity 1 even for uplink traffic.

VI. Related Work

As mentioned earlier, there have been several recent efforts to achieve basestation-level virtualization of the wireless resources [7]–[9], [18]. Authors in [7] designed and implemented the NVS system which is leveraged by NetShare as the link-level scheduler. The same authors also extended NVS to achieve the same goal remotely from a gateway in [8]. Authors in [9] propose to virtualize a LTE eNodeB using a hypervisor. Each entity runs its LTE stack in a virtual machine. The hypervisor allocates spectrum (in units of LTE resource blocks) to the different entities in accordance with some guarantee. Authors in [18] propose a theoretical framework to auction the wireless resources of a basestation to the different service providers and model it as a game theory problem. It relies on fine time-scale (order of 5-10 miliseconds) interactions between the entities and the network owner since their solution is tightly integrated with the basestation MAC scheduler; hence such a framework may be hard to design and implement in practice. These works, however, do not (a) consider network-wide resource allocation (b) fail to effectively distribute the resources of the entities proportional to their demand across the network. Additionally, there are several papers on network sharing for 3G WCDMA networks such as [19]–[21]. However since WCDMA networks are circuit-switched, resource isolation across entities may be achieved by admission control mechanisms. On the other hand, OFDMA based 4G networks such as LTE are packet-switched and as shown via experiments in an earlier section, admission control mechanisms alone fail to provide isolation across entities. Finally, although several GENI [22] design documents describe proposals and issues of wireless network virtualization [23], [24] there are no documents that detail the design and implementation of a system to achieve cellular resource virtualization through spectrum sharing.

VII. Conclusion

Increasing network costs and declining revenues are forcing MNOs to consider RAN sharing in various forms. As network evolve towards being data-dominated from voice-dominated, we believe that RAN sharing will increase service innovation and differentiation. NetShare can be a key technology to realize effective RAN sharing through spectrum sharing. NetShare optimally allocates resources to different entities across a network of basestations, while meeting their reservations. Using large-scale simulations and prototype implementation, we have shown the efficacy of NetShare in realizing this goal.

References

[1] Active RAN Sharing Could Save $60 Billion for Operators. *http://www.cellular-news.com/story/36831.php*.

[2] Telefonica and Vodafone sign sharing deal. *http://tinyurl.com/ahe6kpn*.

[3] Lightsquared: Open-wireless Wholesale Broadband network. *http://www.lightsquared.com*.

[4] Yota Wireless: Russian telcos in LTE network sharing deal. *http://www.yota.ru/en/info/massmedia/details/?ID=284877*.

[5] U. Paul, A.P. Subramanian, M. Buddhikot, and S.R. Das. Understanding Traffic Dynamics in Cellular Data Networks. In *IEEE Infocom*, 2011.

[6] Telecom policy to allow spectrum sharing. *http://tinyurl.com/anjjh6l*.

[7] R. Kokku et.al. NVS: A Substrate for Virtualizing WiMAX Networks. In *ACM MobiCom.*, 2010.

[8] R. Kokku et.al. CellSlice: Cellular Wireless Resource Slicing for Active RAN Sharing. In *Comsnets*, 2013.

[9] L. Zhao et.al. LTE virtualization: From theoretical gain to practical solution. In *ITC*, 2011.

[10] 3GPP. 3rd generation partnership project; technical specification group services and system aspects; network sharing; architecture and functional description (release 9). *3GPP TS 23.251*, V9.4.0, 2011.

[11] F. Kelly. Charging and rate control for elastic traffic. In *European Transactions on Telecommunications*, 1997.

[12] G. Tychogiorgos et.al. Utility-proportional fairness in wireless networks. In *PIMRC*, 2012.

[13] Scott Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communications*, 13:1176–1188, 1995.

[14] OPT++: An Object-Oriented Nonlinear Optimization Library. *https://software.sandia.gov/opt++/*.

[15] Picochip femtocell solutions. *http://www.picochip.com/*.

[16] Intel WiMAX. *http://tinyurl.com/73dp8wz*.

[17] Click modular router. *http://read.cs.ucla.edu/click/*.

[18] Fangwen et.al. Fu. Wireless network virtualization as a sequential auction game. In *INFOCOM*, 2010.

[19] AlQahtani et.al. A Study on Network Sharing and Radio Resource Management in 3G. In *ICTTA*, 2006.

[20] K. Johansson et.al. Radio resource management in roaming based multi-operator WCDMA networks. In *IEEE VTC*, 2004.

[21] M.K. Pereirasamy et.al. Dynamic inter-operator spectrum sharing for UMTS FDD with displaced cellar networks. In *IEEE WCNC*, 2005.

[22] GENI. *http://www.geni.net/*.

[23] D. Raychaudhuri. New Architectures and Disruptive Technologies for the Future Internet. Technical report, GENI Design Doc., 2005.

[24] Sanjoy Paul and Srini Seshan. Virtualization and Slicing of Wireless Networks. GENI Design Document 06-17.

VIII. Appendix: Proof for P_u

To simplify the analysis, we assume a network of B basestations such that each basestation has a unit resource; and let us assume that an entity sets l_j^b at each basestation as $l_j^b = (\alpha \times L_j)/B$

To quantify the loss in utilization for the above formulation of l_j^b, we define the expressions for percentage loss in utilization (P_u) of an entity with α as follows

$$P_u = \frac{(L_j - S_j)}{L_j} \times 100$$

where S_j is the actual allocated aggregate resources to entity j. To derive the expression for P_u for an entity j, we assume the worst case scenario such that the entity has sufficient traffic in L_j basestations to meet its resource reservation of L_j and no traffic at the remaining $(B-L_j)$ basestations. However, NetShare will allocate the minimum resources at every basestation given by $\frac{\alpha \times L_j}{B}$. Hence, the loss in utilization of the entity is the amount of resources that are allocated to the basestations that dont have any traffic to the total reservation L_j of the entity:

$$P_u = \frac{\frac{\alpha \times \ L_j}{B} \times (B - L_j)}{L_j} \times 100$$
$$\implies P_u = \frac{\alpha(B - L_j)}{B} \times 100 \quad (12)$$