

Resource Scheduling For Heterogeneous Traffic in LTE Virtual Networks

Ayman AbdelHamid, Prashant Krishnamurthy, and David Tipper

Graduate Telecommunications and Networking Program

School of Information Sciences, University of Pittsburgh

Email: ata24@pitt.edu, prashk@pitt.edu, dtipper@pitt.edu

Abstract—Cellular network virtualization is being considered as a key trend in future mobile networks towards improved resource utilization. However, virtualization scenarios need investigation to understand the considerations which should be taken into account when deploying virtualized wireless networks in practice. Towards this, we address the performance of a virtualized network in the presence of heterogeneous classes of traffic. In previous cellular network virtualization literature, both Realtime (RT) and Non-Realtime (NRT) traffic requests have been included without distinction. Both types are provisioned using the same algorithm for allocation of resources specified by the Network Scheduler [1]. However, different types of traffic have different characteristics [2], e.g., RT requests are delay sensitive but may need fixed bandwidth, and hence should be treated differently, especially when wireless channel conditions are factored into the scheduling. We recognize this difference and in this paper, we propose a new approach to improve scheduling of resources for RT and NRT traffic. In particular, we prioritize the traffic belonging to different virtual slices from all service providers (SP/VEs) at the Network Scheduler before allocating resources to different SP/VEs, i.e., we form a Virtual Prioritized Slice (VPS). The virtual prioritized slice is forwarded to the VPS scheduler to serve all RT requests first. Only after the RT traffic is scheduled, the NRT traffic is provisioned using proportional fairness (PF) scheduling. We show by simulation results that this new VPS approach helps outperform recently proposed resource allocation schemes.

Index Terms: Heterogeneous Traffic; Real-Time Traffic; LTE Spectrum Virtualization; Cellular Networks.

I. INTRODUCTION

Within the last decade, there has been an enormous increase in the demands of mobile users due to the varieties of services the service providers (SP/VE) have introduced such as video streaming. Hence, spectrum resources are becoming increasingly constrained despite the underutilization of allocated spectrum according to the U.S. Federal Communications Commission (FCC) [3]. Hence, there has been a focus of research in the last two decades that has been directed towards developing new approaches of using spectrum much more efficiently. The idea of Dynamic Spectrum Access (DSA) has been extensively investigated [4], [5]. Moving from underlay DSA [6] towards cognitive radio networks [7] passing through spectrum sharing [8], various types of DSA have been evaluated in the literature towards managing spectrum better.

In a different direction, virtualization is another approach which has been introduced lately to help exploit not only spectrum resources more efficiently, but also to efficiently

share hardware and other core network resources. Virtualization enables the efficient exploitation of physical resources by dynamically assigning resources to service providers subject to their traffic demands and other QoS metrics as shown in Figure 1. In this figure, a certain minimum number of resource blocks in frequency are allocated to each service provider every transmission time interval (TTI). The rest of the resource blocks (RBs) are allocated depending on the requirements of each service provider.

Virtualization showed its promise with the Xen project [9], an x86 virtual machine monitor which allowed multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion. In a similar way, *network virtualization*, defined as decoupling the roles of a traditional network operator into two independent entities, infrastructure providers (InPs) who manage the physical substrate/spectrum and service providers (SPs) or Virtual Entities (VEs) who access different resources of the InP through a virtual network (VN) [10], has shown promise for providing customization, isolation, and resource efficiencies. A Network Scheduler or Resource Manager is an entity that allocates resources to different virtual parties (service providers) according to the network policy/agreement/economics. Virtualization has the benefits of using the hardware resources in more efficient manner, fulfilling excess user demands as well as reducing the CAPEX (capital expenses) and OPEX (operational expenses) burden faced by operators to handle this demand. We use the term SP and VE interchangeably in this paper.

A crucial issue that was investigated in recent literature work in the virtualization context is resource allocation to different service providers or virtual entities operating on the same physical network and sharing spectrum. We use LTE as the cellular system of interest, where resource blocks (RBs) comprising of a subset of frequency channels (sub-carriers) over a given time unit, are allocated to mobiles. Allocating resources to different SP/VEs is a multi-dimensional problem since the Network Scheduler or Resource Manager should assign resources according to each virtual network's demand and the quality of channels in each virtual network, while preserving isolation between different SP/VEs. However, achieving high utility for one SP/VE may affect other entities sharing the same pool of resources. An intuitive proposition for fair resource scheduling among SP/VEs may be to simply apply the well known proportional fairness (PF) algorithm

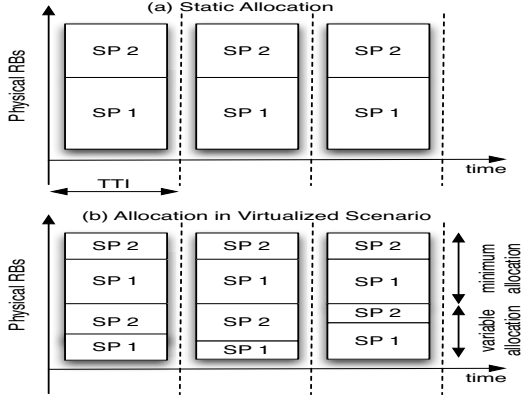


Fig. 1: Allocation of resources in static and virtualized network scenarios

[11] that is used for scheduling mobile users of a single operator in current cellular networks. This algorithm uses the historical throughput that a mobile user has received and the channel conditions to determine whether or not resources are allocated to this user in the current transmission time interval (TTI) (better channel conditions will give a user resources). However, if such a PF algorithm is naively applied in the virtualized case, a possible scenario is that one SP/VE may not be able to acquire resources due to bad channel conditions for his users (see Figure 2). The reason for such a scenario is that the scheduler executes the PF-scheme on users regardless of the SP/VE with whom they are subscribed. This clearly violates the isolation required between SP/VEs and hence the PF scheduling cannot be directly applied *prior to initializing the allocation process*. This problem has been addressed by allocating a minimum guaranteed set of resource blocks (Q_{min}) to each participating SP/VE [12].

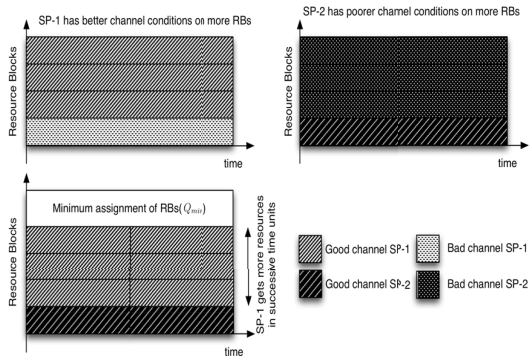


Fig. 2: PF scheduler may allocate more resources to one SP/VE due to channel conditions

In [12], NVS, a resource scheduling framework, is proposed to efficiently allocate resources to different virtual network slices, i.e., a group of flows that belong to a certain SP/VE, subject to its fluctuating demands. Later on, NVS was improved into a scheme called NetShare in [1] where the utility

function which is maximized includes the PF demand for each SP/VE, i.e., it includes both the SP/VE's throughput in preceding time slots as well as the current user demands of an SP/VE, to schedule resources among different SP/VEs (this work assumes SP/VEs share more than one BaseStation (BS)). In both NVS and NetShare, two main assumptions were made – the first is that the utility function does not depend on the possible heterogeneity of user applications (e.g., delay-sensitive applications and elastic applications). As it is well known, the characteristics of real-time communication applications differ significantly from those which are elastic [2]. For instance, in [13], it was shown that 99% of inter-arrival times for VoIP traffic are below 20 ms and the jitter delay is almost always less than 2 ms. Also, real-time traffic is transmitted in small sized packets every constant epoch of time. We can mostly model RT traffic as comprising of constant bit-rate (CBR) traffic. The second assumption in these models is the static assignment of a minimum number of resource blocks Q_{min} , to each SP/VE, as a fraction of the total reserved resources agreed upon in the SLA. As we discuss in the next section, both of these assumptions may lead to inefficiencies in resource allocation.

To the best of our knowledge, preceding work on wireless network virtualization has not examined the implications of including heterogeneous traffic requests in their virtualization framework. Although resource allocation for heterogeneous traffic has been extensively studied in the literature, in a virtualized wireless setting, it poses some interesting challenges. One challenge is ensuring isolation between SP/VEs while serving RT requests since RT traffic is delay sensitive. Another challenge is taking into account the time varying channel conditions in wireless networks. In this paper, we consider these aspects using what we call the Virtual Prioritized Slice (VPS) approach. We argue, by simulation results, that serving the RT requests jointly for all SP/VEs prior to scheduling positively impacts SP/VEs by reducing the user blocking probability for RT flows. While doing this, we consider the allocation of the worst resource blocks (lower data rates) to RT traffic since they do not necessarily need the best throughput. Such an allocation has not been previously considered in the literature.

The rest of paper is organized as follows. In Section II, we provide a background of a virtualized system and motivate the approach taken in this paper through simple examples. In Section III, we introduce our new framework and methods for allocating resources between two SP/VEs with heterogeneous traffic and a dynamic minimum allocation of resources. In Section IV, we present some simulation results of the virtualized LTE network showing how our approach outperforms previously proposed schemes in terms of the total SP/VE throughput.

II. MOTIVATION AND SYSTEM MODEL

In this section, we illustrate the reasons for using the approach examined in this paper. We consider the impact of current scheduling schemes on real-time traffic and how

the currently evaluated schemes show inefficiency in handling real-time requests.

A. Background

In this paper, we consider a single cell of size d_{max} served by one BS (or evolved node B – eNB in LTE) which is physically maintained by a certain InP. We assume two SP/VEs, VE1 and VE2 operating on the same eNB. We assume both VEs have mobile users in the cell with the same coverage. Also, we assume both SP/VEs share the entire spectrum (alternatively, the individual spectrum the 700 MHz-LTE bands are aggregated to form a spectrum pool of bandwidth $B = B_1 + B_2$ where B_1 and B_2 are the bandwidths that are otherwise allocated separately to SP/VE1 and SP/VE2, respectively).

a) *CQI*: Since we assume SP/VEs are operating on an LTE-based physical substrate, physical resource blocks are allocated to users based on the channel quality index (CQI). The CQI informs the eNB about the channel conditions for a given mobile in a given set of RBs. In this paper, we compute CQI according to the table in [14]. The corresponding transmission rate per Hz(η) is derived from Table I¹ below.

TABLE I: CQI and MCS Table

CQI index	modulation(M-ary)	code rate x 1024	efficiency(η)
0	out of range		
1	QPSK	78	0.1523
2	QPSK	120	0.2344
3	QPSK	193	0.3770
4	QPSK	308	0.6016
5	QPSK	449	0.8770
6	QPSK	602	1.1758
7	16QAM	378	1.4766

b) *RT Traffic*: In LTE, RT traffic is assumed to comprise of 1518 bytes packets as indicated in Table I in [15], i.e., 40 bytes payload and 1478 bytes as TCP header overhead. RT frames generated every 20 ms, i.e., the inter-arrival time is 20ms [16]. A fixed number of RBs are sufficient for RT traffic packets. We assume throughout the paper that each RT request is assigned 2 RBs in any given transmission time interval (TTI). We show that 2 RBs are sufficient to provision an RT request by computing the number of transmitted bits per RB using the RB with the lowest CQI. According to table I, the lowest CQI corresponds to $CQI - 1$ with a code rate efficiency of 0.1523 bits/Hz. If we denote the number of bits allocated to an RT request by B_{rt} , we can see that B_{rt} can be computed roughly as:

$$\begin{aligned}
 B_{rt} &= 2 \times 12 \times 15KHz \times 0.1523 \text{ bits/Hz} \\
 &= 54.83 \text{ Kbits} \\
 &\simeq 6853 \text{ Bytes}
 \end{aligned} \tag{1}$$

Recalling that an RT packet size including transport layer overhead in LTE standard is 1518 bytes, it is sufficient that 2 RBs are sufficient to provision a single RT request.

¹This table shows a partial set of CQIs and the complete table is available in [14]

We also note here that since the lowest CQI can still serve RT requests, it will be beneficial to allocate the worst channels to RT traffic and save the best channels, which have higher data rates for non real-time traffic. We explore this later.

B. Problems with Separate Scheduling for SP/VEs

Next we examine why separate scheduling of RT traffic flows *after* allocation of RBs may result in more blocked RT requests through a simple example. In the framework shown in Figure 5, the Network Scheduler is in charge of estimating the number of spectrum resources that should be assigned to each SP/VE based on the proportional fairness allocation scheme. This scheme operates as follows:

$$\begin{aligned}
 &\text{maximize} \quad \sum_g v(R_g) \\
 &\text{subject to} \quad R_g \geq Q_{min}^g \forall g
 \end{aligned} \tag{2}$$

where $v(R_g)$ is the utility function which is maximized to attain the best resource allocation, $R(g)$ is the total transmission rate for VE g and g is the slice number, (for example, slice 1 belongs to VE1, slice 2 belongs to SP/VE2 ... and so on). Also, Q_{min}^g is the minimum guaranteed number of resource blocks needed to satisfy the service level agreement (SLA) for VE- g . According to [17], the proportional fairness utility function can be written in terms of a concave function as follows

$$\begin{aligned}
 v(R_g) &= \varphi_g \log(C_g) \\
 &= \frac{R_g}{T_g} \log(C_g)
 \end{aligned} \tag{3}$$

where C_g is the number of resources acquired by VE- g at the beginning of a TTI and T_g is the aggregate transmission rate obtained by VE- g in all former TTIs starting from $t = 1$ where t is the slot number. The higher T_g is, the lower is the utility. The higher C_g is, the higher is the utility.

Such a log utility function is compatible with NRT traffic. However RT traffic with constant inter-arrival times (as shown in Figure 3) does not observe any improvement in utility with larger bandwidths². In contrast, real-time traffic flows are delay sensitive and should not encounter a delay beyond a maximum acceptable delay. If the delay is larger than this maximum, RT packets are dropped (or flows are blocked). Thus, one of the common methods to handle RT traffic is the notion of priority queues [18]. In LTE, flow requests are initially prioritized before being served according to the criticality of the user application [16], e.g. VoIP(2), Video call(4), ... and so forth. The highest priority flow request (the lowest priority index) is served first.

Unlike traditional LTE networks, in virtualized LTE networks, it may not be the best approach to execute the prioritization step for each SP/VE separately. We claim that prioritization between different traffic types should be performed for both SP/VEs *jointly* at the network scheduler prior to the allocation of resources. The reason for that is that it is

²This may change with different qualities of RT traffic, such as HD voice and video, which is outside the scope of this paper

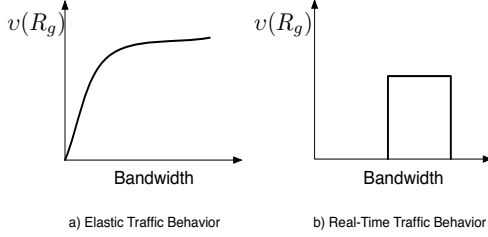


Fig. 3: Utility function versus Bandwidth in NRT and RT traffic cases

possible that a SP/VE would have insufficient resources to serve all real-time requests if the Network Scheduler assigns resources on a fair-throughput basis regardless of the traffic heterogeneity – this is the typical case identified as a problem in this paper.

To clarify this, we use a small example. Suppose we have a total of 8 PRBs and VE1 and VE2 each has 2 RT requests at the beginning of a TTI as shown in Figure 4. We assume also for simplicity that VE1’s RBs all (with transmission rates per Hz) correspond to $CQI_{index} = 11\text{--}64$ QAM and efficiency of 3.32 (not shown in Table I) and similarly, VE2’s RBs transmission rates correspond to $CQI_{index} = 14\text{--}64$ QAM and efficiency of 5.11. Hence, assuming $T_1 = T_2$ for simplicity, from Equation (2), the total utility function per Hz is given as,

$$\begin{aligned}
 U_{total} &= \text{maximize}_{C_1, C_2} v(R_1) + v(R_2) \\
 &= \text{maximize}_{C_1, C_2} R_1 \log C_1 + R_2 \log C_2
 \end{aligned} \quad (4)$$

By solving the simple optimization problem in 4 numerically, we can simply show that the proportionally fair allocation policy in this scenario is $(C_1 = 3, C_2 = 5)$, i.e., these are the numbers of allocated RBs for VE1 and VE2 respectively which achieve the maximum total utility.

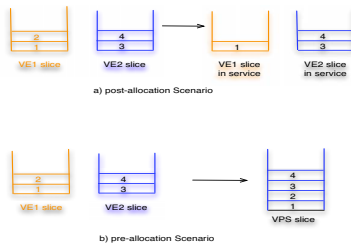


Fig. 4: An example to show the advantage of priority scheduling prior to allocation of resources by the Network Scheduler

We now examine what impact this has on the blocking of RT requests. In Figure 4a, prioritization is carried out individually by each SP/VE after allocation of RBs by the Network Scheduler, i.e., post-allocation priority. As we previously assumed, an RT request is served using 2 RBs. Hence, VE1 would not

be able to serve all of its RT requests and would have one dropped RT request. In the second scenario in Figure 4b, let us suppose that the network scheduler prioritizes requests first and serves them using all of the available RBs. Then, none of the VEs would encounter drops of RT requests. This simple example is just a motivation to argue that the existence of a priority-phase *prior to allocating resources* and then serving RT requests first, independent of the PF-criterion is useful to ensure that that RT requests are better provisioned.

III. MODEL OF PROPOSED SYSTEM

In this section, we describe the model of our proposed approach. Unlike the virtualization framework proposed for NVS [12] and NetShare [1], we adopt a framework that is shown in Figure 5. In this framework, the Network Scheduler performs resource scheduling in two main phases, a priority-phase and a virtual prioritized slice (VPS)/NRT scheduling-phase. First, all the RT flows are scheduled. The remaining resource blocks for NRT traffic are divided between the service providers, ensuring that a minimum number of resource blocks Q_{min} is allocated to a given SP/VE (if available). This number will dynamically change every TTI in a manner similar to proportional fairness. However, due to space limitations, we do not elaborate on the details of how Q_{min} changes in this paper.

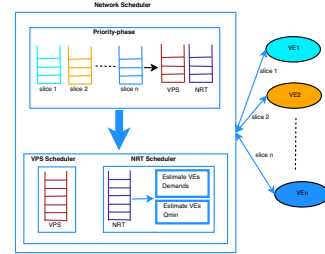


Fig. 5: The new virtualization framework

A. Priority-Phase

In the priority phase, instead of performing prioritization between different classes of applications for each SP/VE separately after allocation of RBs (post-allocation priority), the Network Scheduler collects the flow requests from both entities VE1 and VE2, and jointly prioritizes the flows according to a priority index. This priority index considers the type of application before allocation of RBs (pre-allocation priority). Flow requests from all SP/VEs are queued as shown in Figure 5 into two queues – a virtual prioritized slice (VPS) which contains all RT requests from both SP/VEs and the NRT slice containing all NRT requests from both SP/VEs. The VPS is forwarded to the scheduler to be assigned RBs regardless of the Proportional Fairness scheduling scheme in equation(2). This is because, as we discussed previously, each RT request needs a fixed assignment of RBs (two in this paper). On the other hand, the NRT slice is provisioned using the PF-scheduler in equation(2) according to the number of requests per SP/VE as well as the PRBs quality for each SP/VE.

B. Optimal Network Scheduler

The second phase for the virtual framework comprises of the VPS scheduler and the NRT scheduler which perform scheduling in two steps. In the first step, resources are allocated to RT requests without using proportional fairness. More specifically, RT requests are first allocated 2 RBs each. We examine two approaches later - assigning the RBs with the lowest CQIs and assigning RBs with the highest CQIs. As noted previously, assigning RBs with the highest CQI does not materially improve the performance for RT requests. RT requests are also arranged in order of their delay budgets as explained later. The ones with the smallest delay budgets are allocated RBs earlier. Next, the NRT requests are scheduled according to Equation (2) using the residual RBs.

IV. SIMULATION RESULTS

In this section, we evaluate the performance of the approach proposed in our model, pre-allocation priority (the dynamic nature of Q_{min} is included here, but not discussed). We compared our approach to NVS [12] and NetShare [1]. However, in [1], it was shown by simulations that NetShare outperforms NVS, i.e., it achieves more efficient resource scheduling. Hence, we limit our comparison to NetShare only to provide more clarity to the plots. We conducted simulations using MATLAB. The default parameters used in this simulation are shown in Table II unless otherwise indicated. We assume that all RBs face free space path-loss with uncorrelated Rayleigh fading with other parameters shown in Table II. Our results are calculated based on an average of 5000 iterations per 20 TTIs. Mobile users of both SP/VEs are distributed uniformly in the cell.

TABLE II: Simulation Parameters

Transmitted BS power	10 watts
Cell Size(d_{max})	600 m
Number of active users-SP/VE1(N_1)	20
Number of active users-SP/VE2(N_2)	5
flow length(L)	$3 * 10^5$
SP/VE1-Bandwidth	1.25 MHz
SP/VE2-Bandwidth	1.25 MHz
subcarrier spacing(Δf)	15 KHz
Number of subcarriers/RB	12
subframe(TTI) time	1 msec
Number of symbols/TTI	2

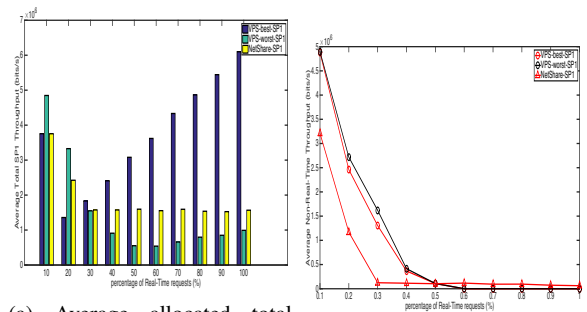
Note that, in simulating the received signal-to-interference and noise ratio (SINR) γ , we assume that the serviced requests always undergo a given constant interference from surrounding cells during their service, for simplicity. The CQI, for determining the throughput, is calculated using a conversion equation based on [19]. The expressions for the received SINR and the path-loss with Rayleigh fading is derived from [20]. We do not include the expressions here due to space limitations.

A. Effectiveness of pre-allocation prioritization at the Network Scheduler

In this part of evaluation, we assume RT requests with exponentially distributed maximum request delay budgets. We

recall that the maximum-request-delay budget is the maximum delay a request can tolerate before it is discarded and dropped. Also, we assume that the inter-arrival time for RT requests is one TTI, i.e., 1 ms. We recall that in our proposed VPS approach, each RT request is serviced by 2 RBs as we presumed antecedently. Figure 6a shows the average total SP/VE throughput for VE1 versus the percentage of RT requests for both the proposed approach VPS (pre-allocation) for two cases – the best (highest) CQIs case and the worst (lowest) CQIs case, and NetShare (post-allocation). It is clear that the VPS approach with worst-CQI outperforms the NetShare and the VPS approach with best-CQI when the network is lightly loaded by RT requests. This is due to the fact that the approach that assigns the worst RBs to RT traffic helps NRT requests with better RBs which results in a higher total throughput. Nevertheless, as the percentage of RT requests increases in the network, i.e., the network becomes more loaded, the best-CQI VPS approach outperforms the worst-CQI approach in allocation since the RT traffic throughput becomes dominant when calculating the overall SP/VE throughput.

Meanwhile, guaranteeing RBs for RT requests does not significantly affect NRT requests since we limit the number of RBs per an RT request to 2 RBs only. To support this statement, Figure 6b shows the average NRT traffic throughput for VE1 versus the percentage of RT requests to the total number of requests. It is apparent that the VPS approach has higher NRT throughput than NetShare. However, as the RT traffic load increases (more than 50%), the three approaches behave roughly the same. This is due to that most of RBs are occupied by RT traffic flows leaving very little for NRT traffic, since RT traffic is prioritized first. Similarly, Figures 7a and 7b show the total throughput as well as the NRT throughput for SP2/VE2.

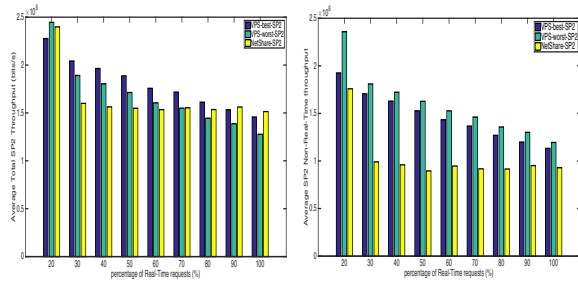


(a) Average allocated total throughput versus the percentage of RT requests (b) The NRT throughput Vs. the percentage of RT requests

Fig. 6: The Average Throughput for SP/VE1 (total and NRT)

V. DISCUSSION AND FUTURE WORK

In this Section, we address some considerations and limitations of this paper which need further investigation. Unlike the static Q_{min} allocation, we have used dynamic Q_{min} allocation. This requires further evaluation and may also require dynamic enforcement of the assigned Q_{min} for both



(a) Average Total throughput versus the percentage of Real-Time requests (b) The Non-real time throughput Vs. the percentage of Real-Time requests

Fig. 7: The Average Throughput for SP/VE2 (total and NRT)

SP/VEs. The enforcement mechanism ensures the compliance of the SLAs between MNOs and SP/VEs over time and acts as evidence against any violations of rules. Such dynamic enforcement may be quite complex and hence costly. Eventually, the assumption of limiting the number of RBs per RT request to 2 RBs only may not be valid in case we consider higher quality of RT traffic, i.e., HD voice and video, especially, if we assumed the lowest CQI-VPS approach. In such cases, the quality of RT traffic should matter and accordingly an RT request should acquire a larger number of RBs or better quality RBs. In Figure 8 we try to show preliminary results on how increasing the number of RBs per RT request will affect the total SP/VE throughput.

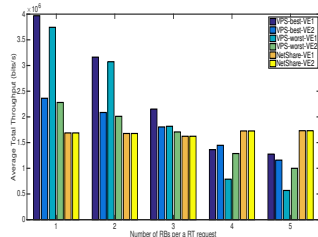


Fig. 8: The Average SP/VE throughput versus the number of RBs per a RT request

VI. CONCLUSION

In this paper, we propose an approach to facilitate heterogeneous traffic in virtualized LTE networks. The reasoning in this work is that real time flows need to be scheduled first, as they need limited numbers of resource blocks, independent of previously achieved throughput or channel conditions. This is in contrast to non-real time flows that stand to gain by being allocated the maximum amount of bandwidth possible. In a virtualized setting where resource blocks are shared between SP/VEs, it is important to have the flexibility in allocating resources. We achieve this in two ways - first, we pool all real-time requests (from all SP/VEs) and schedule them and

second, we allow the dynamic change in the minimum number of resources allocated to a given SP/VE using an algorithm that is similar to the proportional fairness algorithm used for scheduling in LTE networks. We show through simulations that both of these strategies can help improve the performance (reducing the blocking of real-time flows and improving the throughput of non real-time flows).

REFERENCES

- [1] Rajesh Mahindra, Mohammad A Khojastepour, Honghai Zhang, and Sampath Rangarajan, "Nec laboratories america inc., princeton, usa," in *Network Protocols (ICNP), 2013 21st IEEE International Conference on*. IEEE, 2013, pp. 1–10.
- [2] Scott Shenker, "Fundamental design issues for the future internet," *Selected Areas in Communications, IEEE Journal on*, vol. 13, no. 7, pp. 1176–1188, 1995.
- [3] FCC Spectrum Policy Task Force, "Report of the spectrum efficiency working group," 2002.
- [4] William Lehr and Jon Crowcroft, "Managing shared access to a spectrum commons," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*. IEEE, 2005, pp. 420–444.
- [5] Yiping Xing, Rajarathnam Chandramouli, Stefan Mangold, et al., "Dynamic spectrum access in open spectrum wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 3, pp. 626–637, 2006.
- [6] Long Bao Le and Ekram Hossain, "Resource allocation for spectrum underlay in cognitive radio networks," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 12, pp. 5306–5315, 2008.
- [7] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.
- [8] Jon M Peha, "Approaches to spectrum sharing," *Communications Magazine, IEEE*, vol. 43, no. 2, pp. 10–12, 2005.
- [9] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.
- [10] Xin Wang, Prashant Krishnamurthy, and David Tipper, "Wireless network virtualization," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 818–822.
- [11] Paul Bender, Peter Black, Matthew Grob, Roberto Padovani, Nagabhushana Sindhushyana, and Andrew Viterbi, "Cdma/hdr: a bandwidth efficient high speed wireless data service for nomadic users," *Communications Magazine, IEEE*, vol. 38, no. 7, pp. 70–77, 2000.
- [12] Ravi Kokku, Rajesh Mahindra, Honghai Zhang, and Sampath Rangarajan, "Nvs: a virtualization substrate for wimax networks," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 233–244.
- [13] Veljko Pejovic and Heather Zheng, "A real-time traffic packet scheduler for a novel tdma mac protocol," .
- [14] Evolved Universal Terrestrial Radio Access, "Physical layer procedures (release 9), 3gpp," *Technical Specification Group Radio Access Network TS36*, vol. 213, 2010.
- [15] Anna Larmo, Magnus Lindstrom, Michael Meyer, Ghyslain Pelletier, Johan Torsner, and Henning Wiemann, "The lte link-layer design," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 52–59, 2009.
- [16] Erik Dahlman, Stefan Parkvall, and Johan Skold, *4G: LTE/LTE-advanced for mobile broadband*, Academic Press, 2013.
- [17] Frank Kelly, "Charging and rate control for elastic traffic," *European transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.
- [18] Caglan M Aras, James F Kurose, Douglas S Reeves, and Henning Schulzrinne, "Real-time communication in packet-switched networks," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 122–139, 1994.
- [19] Olav Østerbø, "Scheduling and capacity estimation in lte," in *Proceedings of the 23rd International Teletraffic Congress*. International Teletraffic Congress, 2011, pp. 63–70.
- [20] John G Proakis, "Digital communications, 1995," .