An efficient resource management mechanism for network slicing in LTE network

ALFOUDI, Ali, NEWAZ, Shah, OTEBOLAKU, Abayomi <http://orcid.org/0000-0002-4320-9061>, LEE, Gyu Myoung and PEREIRA, Rubem

Available from Sheffield Hallam University Research Archive (SHURA) at:
http://shura.shu.ac.uk/24766/

This document is the author deposited version. You are advised to consult the publisher's version if you wish to cite from it.

Published version


Copyright and re-use policy

See http://shura.shu.ac.uk/information.html
An Efficient Resource Management Mechanism for Network Slicing in a LTE Network

ALI SAEED DAYEM ALFOUDI1,2, S. H. SHAH NEWAZ3,4, ABAYOMI OTEBOLAKU5, GYU MYOUNG LEE2, AND RUBEM PEREIRA2

1Department of Computer Science, College of Computer Science and Information Technology, University of Al-Qadisiyah, Al Diwaniyah, Iraq
2Department of Computer Science, Liverpool John Moores University, Liverpool L3 3AF, U.K.
3School of Computing and Informatics, Universiti Teknologi Brunei (UTB), Gadong BE1410, Brunei
4KAIST Institute for Information Technology Convergence, Daejeon 34141, South Korea
5Department of Computing, Faculty of Science, Technology and Arts, Sheffield S1 2NU, U.K.

Corresponding authors: S. H. Shah Newaz (shah.newaz@utb.edu.bn) and Gyu Myoung Lee (G.M.Lee@ljmu.ac.uk)

This work was supported by Liverpool John Moores University under Grant 10.13039/501100004144.

ABSTRACT The proliferation of mobile devices and user applications has continued to contribute to the humongous volume of data traffic in cellular networks. To surmount this challenge, service and resource providers are looking for alternative mechanisms that can successfully facilitate managing network resources in a more dynamic, predictive, and distributed manner. New concepts of network architectures, such as software-defined network (SDN) and network function virtualization (NFV), have paved the way to move from static to flexible networks. They make networks more flexible (i.e., network providers capable of on-demand provisioning), easily customizable, and cost effective. In this regard, network slicing is emerging as a new technology built on the concepts of the SDN and NFV. It splits a network infrastructure into isolated virtual networks and allows them to manage resources allocation individually based on their requirements and characteristics. Most of the existing solutions for network slicing are computationally expensive because of the length of time they require to estimate the resources required for each isolated slice. In addition, there is no guarantee that the resource allocation is fairly shared among users in a slice. In this paper, we propose a network slicing resource management (NSRM) mechanism to assign the required resources for each slice in a LTE network, taking into consideration the isolation of resources among different slices. In addition, NSRM aims to ensure isolation and fair sharing of distributed bandwidths between users belonging to the same slice. In NSRM, depending on requirements, each slice can be customized (e.g., each can have a different scheduling policy).

INDEX TERMS LTE network, network slicing, wireless virtualization, wireless resource management.

I. INTRODUCTION Today’s network providers contend with the exponential growth of network traffic due to the proliferation of network users and bandwidth-hungry services. The unprecedented growth of mobile networks and the intelligence of smart mobile devices push resource providers to look for more efficient management mechanisms for radio and core network resources in order to improve the users’ Quality of Experience (QoE) and enhance the efficiency of traffic management. According to CISCO, because of the increasing appetite of mobile users for network resources, the mobile network traffic has increased and it is expected to grow to around 70% by 2021 [1], [2]. Taking into account the stupendous growth of traffic, it is timely to redesign the networks in order to meet Quality of Service (QoS) of different applications [3].

To date, many research efforts have been conducted aiming to provide better resource management models in mobile networks (e.g. [4], [5]). Some of these works proposed resource allocation mechanisms based on assigning a number of Physical Resource Blocks (PRBs) to each user’s request in a cellular network. We can broadly classify a resource management mechanism into two levels: a low-level management model and a high-level management model. The advantage of applying a low-level model is that it is easy to implement because any requested resource gets resource allocation in units (e.g. a user in cellular network could get 10 units of PRBs). By utilizing a low-level model, it provides accuracy of allocating resources to each resource demand in units. However, it is hard for the high-level management
entities (e.g. operators and service providers) to adopt a low-
level management mechanism, because resources in the high-
level management model are allocated in portion (e.g. 30% of
total available PRBs).

Looking at the research focus from industries and
academia, we envision that the future network will solely
embrace network virtualization. The major factors that have
resulted in rapid adoption of network virtualization are:
cost-effective sharing of network resources and high net-
work utilization. In order to gain synergistic benefits of net-
work virtualization, along with designing efficient network
architectures, a research effort should focus on an effec-
tive resource management mechanism in a virtual network.
Future virtualized networks need a new management mech-
anism that would provide accuracy of resource allocation
and guaranteed resource isolation. In order to accomplish
these objectives, a novel resource management mechanism
is required that will take into consideration both the low
and high-level management models for resource allocation.
The major role of the low-level model would be providing
PRB based resource allocation in number of units, thereby
ensuring high accuracy in resource allocation. On the other
hand, the high-level model should be capable of ensuring
isolation among the dedicated resources.

In order to facilitate such flexible resource allocation,
dynamic network configuration and cost effective operation
in a network, Software Defined Network (SDN) and Network
Function Virtualization (NFV) open up new opportunities [6].
SDN is an emerging technology where a control plane is
decoupled successfully from a data plane, making a network
programmable and cost effective. SDN offers several advan-
tages over conventional hardware-centric networks, including
on-demand traffic forwarding policy, reduced cost and better
QoS. NFV is a revitalizing technology in future networks.
This allows a physical network infrastructure to be shared
among coexistence of multiple network instances simultane-
ously. SDN and NFV partition the traditional networks into
virtual elements, which are logically linked together [7].

To enable multiple virtual elements to share a common
physical network, the network slicing mechanism comes into
play. Network slicing enables to slice a virtual network across
a Radio Access Network (RAN) and a Core Network (CN).
It is a conceptual architecture that aims to share a common
physical infrastructure among multiple virtual networks using
the same principles applied in SDN and NFV [8], [9]. In par-
ticular, there are some important requirements which should
be met when applying network slicing. These requirements
are summarized as follows:

- Isolation among slices: isolation means the ability of
  restricting the impact of a slice on other slices in the
  same network, even if they share the same infrastructure.
  That is to say, if there is any change of resource status in
  a slice (e.g. traffic load change), such a change should
  not influence the allocated resources of other slices.
- Customization: resource management of each slice
  can be operated independently. That is, the admission
  control policy of a slice can be different from the other
  slices.
- Efficient resources utilization: maximizing the utiliza-
tion of channel resources as much as possible would in
  turn allow increasing the capacity of a base station and
  efficiently utilizing a channel transmission.

Taking into account the aforementioned requirements, for
a LTE network, we propose a Network Slicing Resource
Management (NSRM) mechanism. NSRM aims to ensure
the isolation of allocated resources, fair resource sharing and
customized slice configuration. Most of the existing network
slicing research (e.g. [10]–[12]) demonstrate performance
gain using mathematical analysis. Unlike those research
efforts, in this paper, we evaluate our NSRM in a realistic sim-
ulation environment (we use the OPNET Modeler to simulate
the NSRM proposal). Results obtained through simulation
delineate that NSRM can run different customized traffic
for different slices simultaneously. Additionally, the results
exhibit that, in a LTE network, the solution presented in
this paper can successfully isolate distributed resources of
an eNodeB (base station of a LTE network) among different
slices and increase utilization of network resources.

The rest of the paper is organized as follows. Section II
provides some background information and reviews some
of the existing research on virtual resource allocation using
network slicing. Section III describes the system model and
proposed solution. Section IV provides detailed simulation
results. In Section V, we conclude this paper and present a
future research direction.

II. RELATED WORK

In this section, first we briefly summarize Medium Access
Control (MAC) of a LTE network. Next, we discuss the
existing research efforts in network slicing.

A. MEDIUM ACCESS CONTROL (MAC) IN LTE NETWORK

This sub-section describes the two types of LTE frame struc-
ture. Then, we introduce some of the existing research efforts
in virtualization of network resources in cellular networks.

1) LTE FRAME STRUCTURE

MAC is a layer 2 protocol-stack of a LTE air interface, which
processes the uplink and downlink flows [13]. LTE applies
Orthogonal Frequency Division Multiple Access (OFDMA)
and Single Carrier-Frequency Division Multiple Access
(SC-FDMA) for downlink and uplink communications,
respectively. OFDMA divides the available spectrum into
sub-carriers and allocates these sub-carriers to each user in
the coverage area. The reader is refereed to [14], [15] for
more discussion on the process assigning PRBs and different
scheduling schemes.

2) LTE TRAFFIC SCHEDULING

The LTE standard classifies network services into nine
classes, such that four of them are handled as Guaranteed
Bit Rate (GBR) services, whereas the other five classes are
handled as None Guaranteed Bit Rate (NGBR) services [16]. The LTE scheduler uses these classes to prioritize flow services. An operator sets a scheduling scheme for its eNodeBs. A scheduling scheme should take into consideration different QoS associated with the LTE service class attributes and it is very strict to the priority of flow of services. Due to this strict priority, it would result in either starving of NGBR (best effort) class or in some cases the GBR themselves would face lack of resources when wireless channel condition is less suitable [17].

3) VIRTUAL RESOURCES ALLOCATION IN CELLULAR NETWORKS

We have witnessed many research efforts on wired network virtualization; for example, wired network virtualization for a distributed cloud data center in order to maintain desired Service Level of Agreement (SLA) [18]–[20]. The wired network virtualization is accomplished at different levels of a network such as processor, memory, ports connection and physical link layer. Unlike wired network virtualization, a wireless network requires virtualization in both the CN and RAN. Note that, the concept of wired network virtualization could be applied on the CN. However, accomplishing virtualization in RAN is relatively challenging due to two important reasons: i) a radio link connection is affected by stochastic fluctuation of wireless channel quality, and ii) the wireless networking protocols are completely different from the wired network [21].

In cellular networks, a user may have many flows (user bearers) associated with different applications running at the user’s mobile device. User bearers may share network resources with other bearers of different users through a virtual layer, which is mapped to physical network resources (infrastructure) [22], [23]. In [24], the authors propose a virtual cellular network architecture based on SDN. This architecture facilitates resource virtualization across the CN and RAN for all the packet flows in order to maximize network resources utilization. In their proposals, the authors apply the concept of Virtual Bearer (VB), which has been popularly used in wired networks. The concept of a VB is similar to the PRBs in the LTE architecture. However, there are two basic differences between them. First, they differ in time scale. In case of a PRB, the length of a slot is fixed at 0.5 ms in LTE. On the other hand, in a VB, the length of a slot may be negotiated between the service provider and the network operator depending on requirement(s). Second, in terms of ownership, a VB is owned by a service provider who lacks the knowledge about the wireless resources allocation (a service provider has a concern on meeting QoS requirements of the end users). Whereas, in the case of PRBs, they are owned by a physical Infrastructure Provider (InP).

Next, we introduce the concept of network slicing in brief. Then, we present some of the existing research efforts in network slicing.

B. NETWORK SLICING IN BRIEF

Network slicing is a structure of a virtual network architecture that allows sharing a common physical infrastructure among different virtual networks. It enables a cellular system to share network physical resources residing in CN and RAN among the virtual networks [25]. Figure 1 demonstrates a generic conceptual diagram of network slicing. Generally, cellular networks are composed of two different segments: RAN and CN. However, in the case of network slicing, we need an additional logical functional entity (i.e. Slice pairing function) which facilitates resource mapping between RAN and CN slices, as depicted in this figure. Each network slice is logically composed of one or more Network Functions (NFs) of CN and RAN. Note that, a NF can be occupied by a single slice or shared across multiple slices.

C. EXISTING RESEARCH EFFORTS IN NETWORK SLICING

A large and growing body of literature has investigated architectures for cellular networks slicing. In [10], the authors introduce a Karnaugh-Map algorithm in order to facilitate multiple users access in a virtualized embedded wireless network. This algorithm allows the network to handle real time resource requests. In this work, the authors did not provide an explanation how their proposed mechanism can be implemented in a real hardware, such as in a LTE scheduler.

Authors in [11] extend the work introduced in [10] by considering a case of a dynamic embedded system that rearranges the requests that have already been rejected due to the static nature of the network topology. One major drawback of this mechanism is that its calculation of each scheduling time is too complicated.

The solution proposed in [5] aims to slice the resources of a LTE eNodeB into several virtual networks (slices) so as to allocate each of the slices to different Service Providers (SPs). Each SP has a number of users with different SLAs. The scheduler in an eNodeB assigns a PRB to a user based on the SLA between the user and the SP. For instance, the eNodeB scheduler guarantees that the minimum PRBs that should be allocated to a user. However, it is beyond the scope of this work to ensure isolation among the slices explicitly.
This could result in not ensuring SLAs of all the users. This in turn will result in degrading QoE of some users.

The authors in [31] introduce Network Virtualization Substrate (NVS). The architecture and algorithm of this proposal are designed considering a WiMAX network architecture. The proposal devises a slice scheduler (a slice pairing function), which allows simultaneously coexistence of two kinds of resource allocation mechanisms: resource-based and bandwidth-based reservation mechanisms. In [31], the authors highlight that, flow isolation in WiMAX could be challenging. This is due to the fact that, according to the WiMAX standard, if a flow of a user requires more bandwidth than the initially allocated amount, the scheduler could allow the flow to occupy bandwidth of other flows belonging to the same user. Therefore, in order to ensure flow isolation, the authors propose to modify MAC of WiMAX in their solution. This solution introduced in [31] could be adapted to LTE with some modifications.

A heuristic-based admission control mechanism is proposed in [12]. The proposed idea mainly focuses on prioritization of the slices and users. A RAN scheduler takes into account a user’s satisfaction while scheduling downlink transmission, resulting in improving overall QoE of users. Authors in [12] evaluate their solution based on a mathematical model.

In [26], the authors address the slicing of radio resource allocation among Multi-Tenants where each tenant represents a network operator, and they propose a criterion for dynamic resource allocation based on the weighted proportional fair to achieve the fairness of distributed resources between the tenants and their users.

The authors in [27] consider different traffic classes to forecast on-demand network capacity to accommodate network slice requests based on different SLA, where they are using penalty history and consider one-step training for forecasting error. Unlike the solution provided in [27], we consider a weighted historical value to forecast the resource for each slice which provides more accuracy for resource allocation of slices, also they do not consider the intra slice resource allocation.

In [28], authors introduce a novel network architecture for 5G networks that enables third parties to lease a mobile virtual network from infrastructure providers with the help of a network slice broker. Besides, this architecture provides signaling protocols and interfaces to run a new 5G network slice broker, meaning that the network needs to update the network interfaces to provide admission control and optimize network resources. The research effort in [28] discusses the concept of slice isolation and customization; however, detailed procedures on how to actualize such concept in a 5G network have not been stated. Furthermore, [28] does not provide any performance evaluation results.

In [29], the authors focus on virtualizing the LTE base stations, where the proposed solution (Orion) groups the PRBs convert into virtual Resource Blocks (vRB) groups via a set of abstractions, and supports only relevant information to the corresponding slice. Additionally, it does not consider intra slice isolation and any customization or multiplexing opportunities.

Our previous work [30] introduces a framework showing how LTE and WiFi network can be virtualized. The framework allows both LTE and WiFi networks to slice their network resources and maintain IP-flow mobility for the users. A user can connect with both LTE and WiFi slice when one of them alone is not sufficient to meet QoS requirements. Our work [30] does not present any performance evaluations results of the proposed framework. Furthermore, similar to many other existing works (e.g. [29]), we do not provide any solution for slice customization and intra slice isolation in [30].

The research efforts discussed above are promising. However, they all have one weakness or another. Unlike the existing proposals, the solution we introduce in this paper is not computationally intensive—that is, the solution does not require a long time to estimate resources required in each Transmission Time Interval (TTI). Additionally, in our solution, the user bandwidth request is met with regard to fair sharing of resources among the users belonging to the same slice. It is worth highlighting that our proposed work is capable of optimizing resource allocation in case a slice needs an extra bandwidth in each TTI scheduling time. Finally, it must be noted that most of the existing solutions are evaluated based on mathematical analysis. Unlike the existing solutions, we use the OPNET modeler in order to demonstrate the effectiveness of our solution in realistic scenarios. In Table 1, a qualitative comparison among NSRM and the solutions proposed in [12], [26]–[30] is presented.

### III. PROPOSED NETWORK SLICING RESOURCE MANAGEMENT (NSRM)

This section explains the NSRM solution. NSRM presents three main contributions: (i) a novel architecture framework for virtualizing (network slicing) the LTE network in order to maximize network resources utilization; (ii) a novel algorithm which is capable of dynamically distributing bandwidth among different slices within an eNodeB to maximize resources utilization; and (iii) a Max-Min model that ensures isolation of slice resources across flows and secures a fair share of minimum bandwidth among users. The prime objectives of NSRM are twofold: (i) satisfying the requirements of slices in order to meet the users’ QoE, which in turn will lead to maximize revenue of both InP and a slice owner (e.g. SP); and (ii) meeting QoS requirements for all the flows belonging to the same slice.

Figure 2 illustrates the network slicing concept along with the physical entities in RAN and CN of a LTE network. The physical entities shown in this figure take part in forming all the logical entities of the network slices. We would like to clarify here that, in this paper, we assume the core network slicing approach relies on the solution we proposed earlier in [30].

At this point, we need to highlight that in our solution, a slice owner is responsible for scheduling slice resources.
TABLE 1. Qualitative comparison among some of the recent research efforts on network slicing and proposed NSRM.

<table>
<thead>
<tr>
<th>Solution proposed in</th>
<th>Bandwidth reservation</th>
<th>Inter slice isolation</th>
<th>Intra slice isolation</th>
<th>Slice customization</th>
<th>Dynamically reallocation of released resources</th>
<th>Performance evaluation approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>[12]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>numerical analysis</td>
</tr>
<tr>
<td>[26]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>simulation</td>
</tr>
<tr>
<td>[27]</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>numerical analysis</td>
</tr>
<tr>
<td>[28]</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>none</td>
</tr>
<tr>
<td>[29]</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>prototype implementation</td>
</tr>
<tr>
<td>[30]</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>none</td>
</tr>
<tr>
<td>this paper (i.e. proposed NSRM)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>simulation (using OPNET modeler)</td>
</tr>
</tbody>
</table>

FIGURE 2. LTE physical resources with network slices.

It allocates the required resources to each user’s flows according to a predefined SLA. The following sub-section presents the NSRM system model.

A. NSRM SYSTEM ARCHITECTURAL MODEL

In our NSRM architecture, the network slicing is actualized based on SDN and NFV. As mentioned in our previous work [30], all the LTE core network nodes are hosted in a server and each of these nodes is represented by a VNF. We could represent a slice in the core network as a set of VNFs link together as a chain form [32], [33]. Therefore, a slice resource allocation could be represented as a forwarding graph, which refers to the sequence of executions for different VNFs. The conceptual architecture of the NSRM for the network slicing based LTE network is depicted in Fig. 3. This architecture is broadly segmented into three layers: Slice layer, LTE Slice Controller Manager (LSCM) layer, and Slicer layer. Moreover, the architecture facilitates slicing a virtual network into a number of slices each of which is configured based on the service requirements of an operator.

To present our system model, we consider that in a LTE network there are three slices (slice A, slice B and slice C), as shown in Fig. 3. We consider that each slice belongs to an operator and it is managed by its controller (Slice pairing function). The controller is in charge of maximizing utilization of the slice resources (all the virtual resources).
provide information (policy and resource requirements) to the slice layer in order to assign resources to each of the slices accurately. Next, we provide a detailed explanation on how these elements under each layer function.

1) SLICE LAYER
As we mentioned earlier, each slice in this layer is owned by a slice owner and a slice controller is in charge of managing resources of a slice, as we can notice from Fig. 3. The controller coordinates the interaction among slice elements and stores all slice information, such as users information and resource requirements, in the User Information Database (UID), as depicted in Fig. 4. The following are the main elements of the slice layer:

- User Requests (UR): this element holds user requests. When a user wants to have a service from a slice, first, it needs to invoke the associated UR element of the slice. The user then sends a request message to the slice controller mentioning the service (e.g. video streaming service) it requires. Next, the slice controller determines the amount of required resources (e.g. PRBs) to meet the requirements of the service. Upon receiving this information from the user, the UR stores the information in the UID. The slice controller retrieves user requirements from the UID whenever required.
- User Policy (UP): this element handles a policy for each user (i.e. each user is associated with a policy). The policy is defined by the policy administrator. The slice controller uses the policy defined for a user while processing any requests from the user.
- Resource Computing Per User (RCPU): RCPU computes the resource requirement in order to satisfy the request of a user. The slice controller of a slice uses RCPU to know the exact number of slice resources required to meet a user’s request. The RCPU retrieves a user’s information from the UR and UP before computing the resource requirement for the user.
- User Status: a user could be in an active or idle mode at a given time [35]. This element periodically tracks the status of a user (i.e. active or idle). This facilitates the controller to release the allocated resources of a user if the user is found in idle mode at a given time.

As we mentioned [31], slice resource isolation can be classified into three general categories depending on: (i) group of users with the same type of application, (ii) end-to-end networking (different end-to-end flows), and (iii) resources allocated across different slices (the amount of allocated resource is predefined according to a policy). In our work, we consider that the type (i) and (ii) fall under intra slice isolation. Whereas, the type (iii) requires inter slice isolation.

We assume that a policy administrator (see Fig. 4) negotiates with a SP and settles the contract. Besides, it configures the LSCM layer in order to meet the slice requirements defined in the contract.

The elements of Slice layer, LSCM layer and Slicer layer are presented in Fig. 4. In this figure, these elements are logically interconnected to illustrate the main functionalities of the proposed logical framework architecture. The slice layer is a logical layer where each slice controller manages the intra resources of its slice. A slice controller has knowledge on the amount of resources required in a slice. The slice controller would pass the resource requirement information to the LSCM layer where the SGI element stores all the resource requirements of different slices requested by their controllers. Besides, the policy administrator has a set of suitable policies for all the slices. Therefore, the SGI and policy administrator

FIGURE 3. Conceptual a LTE network slicing architecture.
additional resources for the slice. The slice controller then updates the slice resource allocation in the next TTI.

- **Slice Resource Tracker (SRT):** this element has the global view of the slice resources. It periodically observes overall resource utilization of a slice and notifies the slice controller.

- **Resource Estimation (RE):** this element is responsible for estimating the future expected amount of resources that would be required based on the users’ demand within a slice.

2) LSCM LAYER

In our architecture, the LSCM layer manages the LTE core network (it facilitates communication among the CN entities). Additionally, the LSCM has a global view of network resources requirements. It dynamically monitors the status of the network resources through the statistics of required resources and policies of assigning these resources. The following are the main two elements of this layer:

- **Statistics Gathering Information (SGI):** the task of SGI is to obtain statistics of the resource required for each slice. Periodically, the SGI collects and stores an estimated resource for each slice through the RE element. Therefore, it has a historical statistics of resources for each slice. Based on these statistics, the mean value of required resource is measured in order to realize the exact resource requirement of a slice.

- **Resource Allocation Policy (RAP):** RAP element holds all the policies between the SP and InP. The policy administrator places these polices in RAP. This will allow the Slicer to get policy associated information before allocating resources to each slice (see Fig. 4). Mainly, there are two different categories of slice allocation depending on the type of contract (SLA): Guaranteed bandwidth and Best effort [36]–[38]. We explain them briefly below:

  Guaranteed bandwidth is categorized into two subcategories, as explained below:

  - **Fixed Guaranteed (FG):** in this type of contract, the SP will request the Slicer to allocate a fixed amount of bandwidth all the time (this bandwidth may or may not be 100% utilized).
  
  - **Dynamic Guaranteed (DG):** in this case, the bandwidth allocated to a SP is dynamically changed. The Slicer guarantees bandwidth allocation with the change of a SP’s bandwidth requirement. The SP will pay to the InP depending on the usages.
Similarly, best effort bandwidth is classified into two subcategories, as presented below:

- Best effort (BE) with no guarantee: this type of bandwidth request has less priority than DG and FG. That is, in absence of high priority bandwidth requests (i.e. DG, FG), BE bandwidth request is accepted if the network has available bandwidth.
- BE with Minimum Guarantee (BEMG): in this type of contract, a SP can mention the lower and upper limit of its bandwidth requirement. The Slicer would ensure the lower limit of bandwidth request and the upper limit of a request will be satisfied in presence of abundant bandwidth.

3) SLICER LAYER

As shown in Fig. 4, we introduce a virtual layer (called Slicer layer) on the top of an eNodeB physical resources. The Slicer concept introduced here is similar to the Flowvisor concept, which is designed for wired network virtualization [39].

The Slicer is in charge of virtualizing the eNodeB by creating several virtual eNodeBs where each of this eNodeB represents a network slice. It schedules eNodeB physical resources among slice instances. That is, the Slicer allocates bandwidth resources (PRBs) to each slice using a bandwidth allocation algorithm after taking into account predefined contracts between an SP (slice owner) and InP. Note that it is challenging for the Slicer to allocate PRBs to the slices in a fair manner. To obviate this, in this paper, we come up with an algorithm, which is referred to as a simple exponential smoothing model, to measure the number of PRBs required for each slice (Section III-B.1.c presents this model in details). The following are the main elements of the Slicer layer:

- Virtual Resources (VRs): the task of VRs is to create a logical platform and divide this platform into different logical instances, where each logical instance represents a slice. Moreover, the VRs have two components running the functionality of this platform (see Fig. 4):
  - Per Slice Resource Management (PSRM): PSRM controls a configuration of slice resources between users of a slice. Additionally, PSRM with the slice controller enables distribution slice of resources among the users of slice in a fair manner utilizing the concept of Max-Min model.
  - Resource Computing (RC): RC is responsible of computing the estimated resource of each slice. RC utilizes the exponential smoothing model to calculate required physical resources in PRBs for each slice in every Round Trip Time (RTT). Moreover, SGI and RAP of LSCM layer are providing the RC with required statistics and policy rules to complement a process of slice resource allocation.
- Multiplexing/DeMultiplexing (Mux/DeMux): it is responsible for managing multiple data streams coming from/to different slices over eNodeB channel. Moreover, the Slicer uses this element in order to facilitate mapping between virtual and physical resources (see Slicer layer in Fig 4).

B. NSRM SOLUTION

In this sub-section, we present our NSRM solution. Before we delineate the proposed solution, we present mathematical models which assist the algorithms introduced in NSRM for making a decision in network resources allocation. We devise two mathematical models: the exponential smoothing model and the Max-Min model. The first model has the objective to quantify resource allocation among slices. The second model is formulated with the objective of fair resource allocation among the users in a slice. Our proposed NSRM presents two algorithms: (i) Resource estimation algorithm, which uses the estimation model we derive in this section and (ii) Fair resource sharing algorithm that uses the Max-Min model.

1) MATHEMATICAL MODELS FOR ESTIMATING RESOURCE ALLOCATION OF NETWORK SLICES

The resource allocation for slices using exponential smoothing model is presented. In addition, we provide a solution based on user’s fairness and isolation using Max-Min model. The notations used in the paper is given in Table 2.

### Table 2. Notations used in this paper.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>A set of base stations and each base station denoted as $x$</td>
</tr>
<tr>
<td>$V$</td>
<td>A set of slices and each slice denoted as $v$</td>
</tr>
<tr>
<td>$U$</td>
<td>A set of users and each user denoted as $u$</td>
</tr>
<tr>
<td>$B_x$</td>
<td>The base station spectrum bandwidth</td>
</tr>
<tr>
<td>$\eta_x$</td>
<td>The spectrum bandwidth for user within $x$</td>
</tr>
<tr>
<td>$S$ and $N$</td>
<td>Represents the average signal and noise power</td>
</tr>
<tr>
<td>$I_{ui,x}$</td>
<td>The indication of user associated to $x$</td>
</tr>
<tr>
<td>$\rho_{ui,x}$</td>
<td>The percentage of radio resources allocated to user $u$, by BS $x$</td>
</tr>
<tr>
<td>$R_{ui,x}$</td>
<td>The instantaneous user $u$, data rate</td>
</tr>
<tr>
<td>$\delta_{ui}$</td>
<td>The total number of virtual bearers assigned to a user</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>Observation period</td>
</tr>
<tr>
<td>$\rho_u$</td>
<td>The total user bearer data rate over the period $\Delta T$</td>
</tr>
<tr>
<td>$Q_{ui}^{(v)}$</td>
<td>The actual data rate load of a user bearer in a slice $v$</td>
</tr>
<tr>
<td>$(\rho_{ui})_{+1}$</td>
<td>The next time round of scheduling allocation to a user data rate</td>
</tr>
<tr>
<td>$v_B$</td>
<td>A slice bandwidth capacity over base station $x$</td>
</tr>
<tr>
<td>$V_B$</td>
<td>The total slice bandwidth in the base station $x$</td>
</tr>
<tr>
<td>$\lambda_{ui+1}$</td>
<td>The estimate PRBs of a slice during the $(t+1)$ interval time</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>A smoothing constant</td>
</tr>
<tr>
<td>$FF_v$</td>
<td>The fairness factor of a slice $v$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>The total estimated bandwidth of all slices</td>
</tr>
<tr>
<td>$\psi_v$</td>
<td>The total number of PRBs allocated for each slice $v$</td>
</tr>
<tr>
<td>$n$</td>
<td>The number of users in a slice $v$</td>
</tr>
<tr>
<td>$x$</td>
<td>Represents the excess bandwidth for individual user $u$ in a slice $v$</td>
</tr>
</tbody>
</table>

### a: LTE NETWORK VIRTUALIZATION

In LTE, the RAN consists of a number of Base Stations (BSs). Let $X = (x_1, x_2, ..., x_n)$ denote as a set of BSs. We would highlight that, in our propose solution we consider individual BS ($x$) to show the strength of our solution in terms of allocating different slices in one corresponding physical BS, therefore, for each base station $x$ there is a set of slices $V = (v_1, v_2, v_n)$ with a set of users $U = (u_1, u_2, u_3)$ for each $v$. In BS, the spectrum bandwidth allocated to $x$ is $B_x$.
(as described in Section II-A.1). By using Shannon bound, we can define the spectrum bandwidth efficiently for user $u_t$ associated with BS $x$ as shown in (1) [40].

$$
\eta_{ux} = \log_2 \left( 1 + \frac{S}{N} \right),
$$

where $S$ and $N$ represent the average signal and noise power, respectively.

Let $L_{ux}(t)$ be a pointer that indicates the user $u_t$ is associated with BS $x$ or not, where if the $L_{ux}(t) = 1$ means $u_t$ is connected to BS $x$; otherwise $L_{ux}(t) = 0$ means it is not. $Y_{ux}(t)$ represents the percentage of radio resources allocated to user $u_t$ by BS $x$, where $Y_{ux} \in [0, 1]$ and notes that:

$$
\sum_{x_i \in X, v_i \in V, u_i \in U} Y_{ux} \leq 1.
$$

Such that, the instantaneous user $u_t$ data rate is defined by:

$$
R_{ux} = \sum_{x \in x} L_{ux} B_X Y_{ux} \eta_{ux}.
$$

**b: RESOURCES SLICING**

Usually the PRB is assigned to a bearer as a pair of sub-frame in the time domain (described in Section II-A). Thus, we consider one Virtual Bearer (VB) to be equal to pair of PRBs sub-frames representing the resources of a slice in Slicer. Let $\delta_{u_t}$ represents the total number of VBs that the Slicer actually assigns to a user bearer $u_t$ over some observation period $T$. Therefore, the total user bearer data rate $\rho_{u_t}$ over this period is given as illustrated in (4).

$$
\rho_{u_t} = \frac{\delta_{u_t}}{\Delta T}.
$$

Thus, we can formulate the actual data rate load ($Q_{ux}$) of a user bearer in a slice over a base station from (3) and (4):

$$
Q_{ux} = \rho_{u_t} R_{ux}.
$$

The slice has to allocate and prepare required resources by the Slicer to satisfy a user data rate each time trip as shown in (6):

$$
(\rho_{u_t})_{t+1} \geq (\rho_{u_t})_t.
$$

At least the minimum value of the current $(\rho_{u_t})_t$ total user bearer data rate $\rho_{u_t}$ at a time $t$ is required in the next $t$ time trip $(\rho_{u_t})_{t+1}$ of scheduling allocation to satisfy the requirements of a user data rate. Notice that, sometime user data rate in $(\rho_{u_t})_{t+1}$ is greater than the user data rate in $(\rho_{u_t})_t$ to satisfy the user demands (described in Section III-B.1.d).

Therefore, the $(v_B)_t$ is the total slice bandwidth capacity $(v_B)_t$ at a time $t$ over the base station $x$ is:

$$
(v_B)_t = \sum_{v_i \in V} (\rho_{u_t})_t.
$$

Therefore, the total slices bandwidth $(V_B)_t$ at a time $t$ in the base station $x$ is:

$$
(V_B)_t = \sum_{v_i \in V} (v_B)_t, \text{ where } (V_B)_t \leq (B_x)_t.
$$

The $(B_x)_t$ represents bandwidth capacity $B$ for base station $x$ at a time $t$.

**c: SLICER’s RESOURCE ALLOCATION USING EXPONENTIAL SMOOTHING MODEL**

PRBs in a BS need to be allocated and shared between slices based on resource requirements of each slice (as shown in Fig. 4). Thus, each slice should provide an estimated value of the required resources and periodically send them to the Slicer. In order to achieve this, a slice controller needs to calculate required bandwidth of the slice periodically as shown in (7). The LTE Slice Controller Manager (LSCM) collects all estimated bandwidth values from slices and sends them to the Slicer. The Slicer uses these values to allocate PRBs for each slice efficiently. To enable this, we utilize the simple exponential smoothing model as shown in (10).

$$
\lambda_{t+1} = \alpha \times v_{Bt} + (1 - \alpha) \times \lambda_t,
$$

where $\lambda_{t+1}$ indicates the estimate of PRBs for each slice during the $(t+1)$ interval time. The $\lambda_{t+1}$ describes slice status where it either requires additional PRBs or the slice needs to release some PRBs. $\lambda_t$ refers to the current estimate amount of PRBs during $TTI$ interval. $t$ is the Slicer interval, which consists of a number of $TTIs$. $\alpha$ is a smoothing constant, which serves as a weighting factor. Taking consideration $\alpha$, we reformulate (9) as follows:

$$
\lambda_{t+1} = \alpha v_{Bt} + (1 - \alpha) v_{Bt-1} + (1 - \alpha)^2 v_{Bt-2} + \ldots + (1 - \alpha)^{t-1} v_{B1} + (1 - \alpha)^{t} \lambda_1,
$$

where $\lambda_1$ represents a simple average of the $\sum_{i=1}^{n} v_{Bt}$, and $\alpha$ has a value between (0 and 1) where $0 < \alpha < 1$. In (10), too large value of $\alpha$ would result in making value of $(1 - \alpha)^t$ close to zero.

Generally, $\lambda_{t+1}$ has either positive or negative values when compared with $\lambda_{t}$. In the case of a positive value, the slice needs more PRBs, whereas in the case of a negative value, the slice operator satisfies the current state of allocation PRBs. The Slicer utilizes these values to calculate and allocate PRBs to each slice (virtual network). Moreover, this type of calculation is especially useful for network slicing within a contract from type DG, BE or BEMG. The DG contract represents the actual allocated bandwidth to slice operator for serving users requirements, and the maximum bandwidth by the terms of contract. With types BE and BEMG contracts, the Slicer determines the minimum requirements for the BEMG slice operator and the remaining PRBs will be assigned to BE contracts.

The isolation between slices is based on the fairness factor as calculated in the following (11):

$$
FF_v = (\lambda_{t+1})_t / \omega.
$$

$FF_v$ is the fairness factor of slice $v$; $(\lambda_{t+1})_t$ is the estimation value of PRBs for slice $v$; $\omega$ is a total PRBs over all BE slices.
The $\omega$ is computed using the following equation.

$$\omega = \sum_{v=1}^{\text{VBE x slices}} (\lambda_{t+1})_v.$$  \hfill (12)

The total number of PRBs ($\varphi$) allocated for each BE slice $v$ is described as in (13).

$$\varphi_v = \text{int}((FF_v*\gamma)).$$  \hfill (13)

where the $\gamma$ is the remaining PRBs after allocating guaranteed bandwidth to slices.

d: MAX-MIN MODEL FOR USERS FAIRNESS AND ISOLATION IN SLICE

Generally, the scheduling mechanism should be fair and it should isolate the bandwidth between users in the same slice. To realize this, we use the Max-Min fairness model. The Max-Min fairness means maximizing the minimum fair share of the bandwidth for each user within a certain slice. Three principal steps have to be considered in the Max-Min mechanism:

1. Resource allocation is in increasing order of their demands.
2. No user gets a share larger than its demands.
3. Users with unsatisfied demands get equal shares.

Let $U_p$ be a set of users $U$ with their bandwidth demands $p$ in $v$ such that these users are arranged in ascending order, which we formally define as follows:

$$U_p = \{\rho_1, \rho_2, \ldots, \rho_N\}, \text{ such that } \rho_1 < \rho_2 \ldots < \rho_N.$$  \hfill (14)

To equally share slice’s resources (bandwidth) between users, let’s consider $u_E$ is the bandwidth share of individual user $u$ in slice $v$. $u_E$ gets as follows:

$$u_E = \frac{v_b}{n},$$  \hfill (15)

where $v_b$ is the total bandwidth of a slice $v$ and $n$ is the number of users in $v$. So that, the user will be protected by allocating the same bandwidth as other users. Not only that, allocated bandwidth represents the minimum satisfied requirement of a user service in slice $v$.

In some cases, the user’s demands $\rho$ is greater than the allocated bandwidth $u_E$, which means that the user is unsatisfied. In such a case, for all unsatisfied users, they will get the same (equally) extra bandwidth from the slice controller if it is available. In the slice, not all the users are unsatisfied. Some of them have more bandwidth than they actually need. Therefore, we can calculate the excess bandwidth and equally distribute it between unsatisfied users. Thus, assume that $z$ represents the excess bandwidth for an individual user $u$, we compute the value of $z$ as illustrated in (16):

$$z = u_E - \rho.$$ \hfill (16)

Now, for each unsatisfied user in slice $v$, it will get $z/x$ bandwidth, if we assume that $x$ represents the number of unsatisfied users in $v$. The slice operator repeats this process by the slice controller each time if excess bandwidth is available. As a result, no users will get more allocated resources (bandwidth) than they need.

2) NSRM ALGORITHMS

From the previous discussion on how the estimated resource model and the Max-Min model influence the resource allocation, we conclude that both models work in different tiers (intra, inter). In inter-tier resource allocation, the estimated resources are allocated among different slices, whereas the intra-tier resource allocation is a process in which the resources of a slice are allocated among different users in the slice. Here, we propose two algorithms for resource allocation namely, NSRM inter-tier resource allocation (Algorithm 1) and NSRM intra-tier resource allocation (Algorithm 2). Both algorithms are implemented in the Slicer.

Algorithm 1 NSRM Inter-Tier Resource Allocation

**INPUT:** $V$, $B_x$, /*set of slices in a base station*/

**OUTPUT:** $(\lambda_{t+1})_v$. /*PRBs for each slice within the base station*/

**for all** $v = 1$ to $V$ **do**

$$\omega_v = \omega_{v-1} + \text{call} (\text{GET-PRBs})_v/*$$

**end for**

**for all** $v = 1$ to $V$ **do**

$$(\lambda_{t+1})_v/\omega_v$$

**end for**

**if** $\omega_v \leq B_x$ **then**

$v = 0$

**else**

$$(\lambda_t)_v/\omega_v$$

**end if**

**if** $(\lambda_t)_v > (\lambda_{t+1})_v$ **then**

release $\text{PRBs} = (\lambda_t)_v - (\lambda_{t+1})_v$

**end if**

Algorithm 2 NSRM Intra-Tier Resource Allocation

**INPUT:** $\alpha$, $v_b$, $\lambda_t$

**OUTPUT:** return value of $(\lambda_{t+1})_v$ for the calling function.

/* Using (10) calculate $(\lambda_{t+1})_v$ */

/* Where $v_b$ is calculated using (7) */

As mentioned earlier, Algorithm 1 allocates resources among different slices. For that, it needs the required resource of each slice ($v_b$) and the total PRBs of an eNodeB ($B_x$). The algorithm invokes the GET-PRBs function to calculate the estimated resources of each slice according to (10). Then, it finds a value of the total estimated resources of all slices. This algorithm checks whether the total value of slices is less than or equal the total PRBs of the eNodeB. If so, the algorithm assigns a required resource to each slice, otherwise, all the slices continue with the same currently allocated resources until more resources are available in the Slicer. That is, sometimes the estimated forecasting of resource allocation of a slice is less than the current resource allocation. In such
a case, Algorithm 1 will release the surplus resources to allocate to other slices that are unsatisfied with a current resource allocation.

**Algorithm 2 NSRM Intra-Tier Resource Allocation**

**INPUT:** \((\lambda_{i+1} + 1)_v, N, U_p \) \# set of users demand \(p\) in a slice

**OUTPUT:** \(u_E \) \# the bandwidth for each user in a slice

\[ v_b = (\lambda_{i+1} + 1)_v \] \# resource allocation for a slice \(v\) by the Slicer

\[ u_E = v_b / N \]

\[ x = 0 \]

for all \(i\) = 1 to \(N\) do

if \(p_i > u_E\), then

\[ U_E [x] = u_E / \# all unsatisfied users will store in \(U_E\) set \#\]

\[ X = X + 1 \]

either if

\[ i = 0 \]

while \(u_E_i > p_i\) do

\[ z = u_E_i - p_i \]

\[ z/x \# all the users in \(U_E\) get z/x share resources \#\]

\[ i = i + 1 \]

either end while

In Slicer, the Algorithm 2 is responsible for intra-tier resources allocation. This algorithm requires to know the number of users \(N\) in the slice along with their resource demands \(U_p\) and the overall resources allocated to the slice \((\lambda_{i+1} + 1)_v\) from the Slicer.

According to the Algorithm 2, initially, all \(N\) users get equal share of resource \(u_E\). Then, the algorithm checks whether a user demand \(p_i\) is greater than \(u_E\) or not. If \(p_i\) is greater than \(u_E\), i.e. the assign resource for a user is unsatisfied, the algorithm will add the user to a list of unsatisfied users. This process will continue until all users are checked. Moreover, the algorithm will check if there is any user whose \(u_E\) is greater than \(p_i\). If so, this will distribute equally the surplus resources from the user among the all users in the unsatisfied list. This process continues until all the users in the slice are checked. Following these processes mentioned above, the algorithm meets demand of resources of all the users as much as possible.

**IV. PERFORMANCE EVALUATION**

This section is divided into two parts: the simulation configurations and simulation results. The configuration of the simulation explains the topology of the network used in the simulation. In the second part, we present the simulation results and explain the significance of our results. For validation purposes, we evaluated our proposed solution in different scenarios as presented in the next sub-sections.
sub-section IV-B.3 showing how effectively each of the slices can be customized under the proposed NSRM.

1) BANDWIDTH RESERVATION
This sub-section presents different scenarios of bandwidth reservation based predefined contracts of slices with an InP as follows:

For the fixed guaranteed bandwidth contract, we consider a video traffic model. In this scenario, we assume that the downlink (DL) of an eNodeB provides 30 PRBs (a fixed guaranteed user data rate).

Figure 6 shows the average user throughput under a legacy LTE network and the proposed NSRM. As depicted in the figure, both networks show approximately the same per-user throughput performance. This happens because in the case of fixed guaranteed bandwidth both solutions follow the same mechanism, as we mentioned before. Unsurprisingly, due to the same reason, both of the solutions present similar average end-to-end delay performance (see Fig. 7).

The next scenario is based on a dynamic guaranteed bandwidth contract with the VoIP traffic model application. In this scenario, the DL user data rate dynamically changes based on users’ requirement and the maximum guaranteed boundary of resource reservation is 30 PRBs. Figure 8 demonstrates throughput performance comparison between these two solutions. The result shows that the average throughput per user in both networks is similar. The reason for this is that under both solutions the bandwidth reservation is guaranteed even with dynamic changes of user throughput. This scenario proves that the NSRM solution is able to dynamically reserve PRBs of a slice according to users’ requirements.
At this point, we are interested to observe how the proposed solution can contribute in maximizing utilization of radio resources. Figure 9 demonstrates resource blocking performance comparison between the two solutions. The results depicted in this figure confirms that in NSRM resource blocking is approximately 35% less compared to the legacy LTE network. The rationale for this result is that unlike the legacy LTE (see LTE bandwidth allocation mechanism in Section II-A.2), our proposed NSRM allocates bandwidth based on slice requirement, resulting increasing utilization of PRBs (i.e. there would not be any unused PRBs). Consequently, in NSRM, the resource blocking would be less compared to the legacy LTE network.

In addition, we are interested in observing the importance of the proposed solution when a network has best effort traffic. In our simulation, in this case, we consider three types of traffic: best effort, guaranteed bandwidth and dynamic guaranteed bandwidth. Traffic of VoIP and video application services is considered as best effort in our simulation. Both of these applications have a minimum and maximum guaranteed data rates of 30 PRBs and 50 PRBs, respectively.

Figure 10 shows the average bandwidth of VoIP service per-user in a legacy network and NSRM solution. In this figure, we can note that both networks have the same performance per user bandwidth. Note that both networks assign the remaining PRBs to the best effort applications after satisfying resource demand of the guaranteed bandwidth applications. In case of VoIP traffic, both solutions can meet the bandwidth requirement. Consequently, their performance for VoIP service is the same. However, the results for average bandwidth allocation for a video service depicted in Fig. 11, show that the NSRM outperforms the legacy LTE network. It needs to highlight that VoIP traffic is given more priority than the video traffic in a LTE network [14]. Therefore, after meeting the VoIP traffic bandwidth requirement, the legacy network allocates the residual bandwidth to the video services. The NSRM does the same; however, the amount of residual bandwidth in NSRM is larger than a legacy LTE network due to applying the dynamic bandwidth allocation mechanism. Consequently, in NSRM, a user gets more bandwidth compared to a user in a legacy LTE network (a user approximately gets 15 Kbps more bandwidth in NSRM).

2) EVALUATION OF ISOLATION MODEL

In this section, we demonstrate how our solution can successfully maintain the isolation for both inter slice (among the slices) and intra slice (among the users belong to the same slice). Under the same scenario, we compare NSRM’s results in front of a legacy network. In this simulation scenario, we consider FTP traffic flows. Here, we consider two groups of users. The first group (slice 1) and the second group (slice 2) has 5 users and 3 users, respectively.
All the users in our simulation are located at equal distance from an eNodeB, which applies 64 QAM for Modulation and Coding Schemes (MCS). Furthermore, we assume as an aggregation, bandwidth requirement is 9 Mbps and each of the slices needs 4.5 Mbps. Additionally, in this performance evaluation, we assume all the users in a slice have the same bandwidth requirements. Simulation results are presented in Fig. 12 (a) and (b) for a legacy network and NSRM, respectively.

Looking at Fig. 12 (b), we observe that NSRM successfully isolates resources between the two slices. That is, NSRM provides both of the slices an equal amount of bandwidth (each slice gets 4.5 Mbps). From the same figure, we can also realize that, under each slice all the users are provided almost the same amount of bandwidth. These results clearly highlight that NSRM can successfully isolate not only the inter slice bandwidth but also it can isolate users’ bandwidth within a slice (e.g. in case of slice 1, each of the five users gets around 0.9 Mbps).

In the next simulation scenario, we aim to illustrate how our proposed NSRM can dynamically reallocate bandwidth and successfully isolate resources with the change of network conditions. We narrate the scenario as follows. In this case, our assumptions are the same as the previous scenario. Further, in this simulation, we consider, initially, each of the 8 users connected with an eNodeB is allocated 1.125 Mbps (i.e. the eNodeB provides total 9 Mbps to these users). After 200s from the simulation starting time, two users (users 6 and 7 in Legacy LTE, and 1 and 2 of Slice 2 in NSRM) turn off their mobile, releasing around 2.25 Mbps bandwidth in each scenario. In case of Legacy LTE, the scheduler will redistribute the released bandwidth equally to the remaining users. However, for the NSRM, the slice controller (scheduler) of the slice will reallocate the released resources of the slice and distribute them to the users according to their current requirements. The simulation results from this scenario are presented in Fig. 13 (a) and (b).

Looking at Fig. 13 (a), we observe that in a legacy network overall bandwidth of each user is increased by 0.375 kbps after two users left the network (see Fig. 12 (a)). It happens because, in a legacy network, the eNodeB redistributes the released bandwidth across the users equally. In case of NSRM, as we notice from Fig. 13 (b), the user 3 of Slice 2 is reallocated the released bandwidth (See Fig. 13 (b)). However, the bandwidth allocated to each user in Slice 1...
remains the same (i.e. the change of bandwidth allocation in Slice 2 does not influence the users of Slice 1). This result clearly proves that NSRM does not only successfully isolate resources between the slices but that it can also dynamically reallocate the resources.

3) CUSTOMIZATION

In this sub-section, we want to demonstrate that in our NSRM each slice can have its own scheduling policy (i.e. different slices can have different scheduling policies). Let us assume that Slice 1 and Slice 2 each has 4 users with heavy video traffic flows. In this simulation scenario, we consider that the Slice 1 uses a Priority Round Robin (PRR) and the Slice 2 applies Weighted Fair (WF) scheduling policy. Moreover, we suppose that all users in both slices have the same configuration setup (see video traffic model in Table 3). Simulation results are presented in Fig. 14 (a) and (b) for traffic delay and DL traffic received.

Figure 14 (a) shows the delay performance for each slice in NSRM. From this figure, we can realize that despite having the same number of users with the same configuration in both slices, their delay performances are not identical. In fact, this result is quite obvious. As these two slices have two different scheduling policies, their delay performance is not the same. And due to some reasons, they have different downlink throughput performances, see Fig. 14 (b). Therefore, these findings delineate that the proposed NSRM can allow dispensing different scheduling policies for each of the slices in an eNodeB. Note that the explanation of the performance of these two scheduling polices is not in the scope of this paper.

V. CONCLUSION AND FUTURE WORKS

In this paper, a network slicing mechanism for resource allocation in LTE networks has been presented. The proposed mechanism is based on a simple exponential smoothing model that takes into consideration the estimated bandwidth that each slice needs periodically. In addition, we propose a Max-Min fairness mechanism for isolating and fair sharing of a distributed bandwidth between users. Our simulation results show that the proposed mechanism satisfied the user service requirements and that it can implement different customized flow traffic for different isolated slices simultaneously.

In our future research, we are aiming at investigating how network slicing can be actualized in order to share resources from different heterogeneous access networks (develop a unified network slicing platform). In this unified network slicing platform, among the important resources issues, we are planning to study: (i) QoS aware mobility management, and (ii) energy efficient dynamic network slice selection for user devices.

REFERENCES

ABAYOMI OTEBOLAKU received the Ph.D. degree in telecommunication engineering from the Faculty of Engineering, University of Porto, Portugal, in 2015. From 2009 to 2015, he was a Research Engineer with the Centre for Telecommunications and Multimedia, INESC TEC (formerly INESC Porto). He was a Postdoctoral Research Associate with the Department of Electronics, Telecommunications and Informatics, University of Aveiro, and the Institute of Telecommunications, Aveiro, Portugal. As a Postdoctoral Research Associate, he was with the Department of Computer Science, Faculty of Engineering and Technology, Liverpool John Moores University, Liverpool, U.K. He is currently a Lecturer and a Module Leader in software engineering with the Department of Computing, Sheffield Hallam University, Sheffield, U.K. With participation in several research projects, including European projects, his research focuses on context awareness, activity context recognition, mobile data management, and the IoT-driven personalized services in pervasive and ubiquitous environments. He received the INESC TEC grants for doctoral research, from 2009 to 2011. From 2011 to 2015, he received the Portuguese government Fundacao para Ciencia e a Tecnologia-Foundation for Science and Technology (FCT) full doctoral grants (Bolsa de Deutoramento).

GYU MYOUNG LEE was a Visiting Researcher with the University of Melbourne, Australia, in 2002. He was a Research Professor with KAIST and a Guest Researcher with the National Institute of Standards and Technology (NIST), USA, in 2007. He had been with the Institut Mines-Telecom, Télécom SudParis, since 2008. Until 2012, he was invited to work with the Electronics and Telecommunications Research Institute (ETRI), South Korea. He has been an Adjunct Professor with the KAIST Institute for IT convergence, Daejeon, South Korea, since 2012. He joined the Department of Computer Science, Liverpool John Moores University (LJMU), U.K., in 2014, as a Senior Lecturer, where he was promoted to a Reader, in 2017. He also has work experience in industries in South Korea. He has contributed more than 400 proposals for standards and published more than 160 papers in academic journals and conferences. His research interests include the Internet of Things, web of things, computational trust, knowledge centric networking and services considering all vertical services, smart grid, energy saving networks, cloud-based big data analytics platform, and multimedia networking and services. He has been actively participating in standardization meetings including ITU-T SG 13 (Future Networks and cloud) and SG20 (IoT and smart cities and communities), IETF and oneM2M, and so on and currently serves as a Rapporteur of Q16/13 (Knowledge centric trustworthy networking and services) and Q420 (e/Smart services, applications and supporting platforms) in ITU-T. He received several best paper awards in international and domestic conferences. He is also the Chair of the ITU-T Focus Group on Data Processing and Management (FG-DPM) to support the IoT and smart cities and communities. He also served as a Reviewer for the IEEE journals/conference papers and an Organizer/Member of the committee of international conferences.

RUBEM PEREIRA received the B.Sc. degree in electronic engineering from Pontifical Catholic University, Rio de Janeiro, the master’s degree (Hons.) in information systems engineering from South Bank University, and the Ph.D. degree from Manchester Metropolitan University, with a focus on the performance analysis and evaluation of computer networks. He is currently a Reader in multimedia computing and the Programme Leader for the Master Programmes with the Department of Computer Science, Liverpool John Moores University (LJMU), U.K.