Resource Allocation with Admission Control for GBR and Delay QoS in 5G Network Slices

Tulja Vamshi Kiran Buyakar
Department of CSE
IIT Hyderabad, India
cs16mtech11020@iith.ac.in

Harsh Agarwal
Department of CSE
IIT Hyderabad, India
cs15btech11019@iith.ac.in

Bheemarjuna Reddy Tamma
Department of CSE
IIT Hyderabad, India
tbr@iith.ac.in

Antony Franklin A
Department of CSE
IIT Hyderabad, India
antony.franklin@iith.ac.in

***Abstract*—Network slicing is an integral part of 5G which supports next-generation wireless applications over a shared network infrastructure. It paves the way to leverage the full potential of 5G by increasing the efficiencies through differentiation and faster time-to-market. In this work, we propose a Mobile Virtual Network Operator (MVNO) Slice Resource Allocation Architecture (MSRAA) for supporting different network slices in the 5G data plane. MSRAA supports QoS parameters, including Guaranteed Bit Rate (GBR) and Maximum Delay Budget. Using long short-term memory (LSTM) neural networks, we predict network slices bandwidth requirements for efficiently allocating the resources. To reduce revenue loss to the network operators due to forecasting errors, in the proposed Bandwidth Admission Control (BAC) algorithm reallocates resources from lower priority slices (ex: guaranteed service users) to higher priority slices (ex: best-effort users). Using Mondrain Random Forests in our Delay Admission Control (DAC) algorithm, we predict the end-to-end delay and admit flows into slices that can satisfy delay requirements.**

We implement MSRAA on our developed 5G Core testbed and evaluate User Service Request (USR) acceptances and do a complete cost-benefit analysis of our architecture. We show that for eMBB-GBR and eMBB-Non-GBR slices, our algorithm is showing a significant reduction in costs and an increase in profits.

I. Introduction

The upcoming 5G is going to have a substantial influence on nearly every facet of life. The network services deployed in 5G will have a wide range of verticals and use cases. The enhanced Mobile Broadband (eMBB) services in 5G aims to focus on services that require high bandwidth and sustained high capacity network connections, such as virtual reality (VR), high definition (HD) videos, etc. 3GPP has defined various Quality of Service (QoS) characteristics in [1], some of which are mentioned in Table I. As shown in Table I, these services have QoS in terms of Guaranteed Bit Rate (GBR) or Non-Guaranteed Bit Rate (NGBR), packet delay budget and packet error rate.

The key to the evolution of 5G networks is end-to-end (E2E) Network Slicing, which is crucial for supporting diversified 5G services. Network slicing is a specific form of virtualization that allows multiple logical networks to run on top of a shared physical network infrastructure. With Software-Defined Networking (SDN) as an underlay and Network Functions Virtualization (NFV) supporting the underlying physical infrastructure, 5G will "cloudify" radio access and packet core elements. Efficient inter-slice Management and Orchestration (MANO) plays a vital role in 5G networks. To guarantee the QoS requirements for every flow, the cross-slice MANO needs an advanced policy that optimally chooses to either accept or reject a new User Service Request (USR) according to dynamic resource load and then allocate & orchestrate resources for the USR.

TABLE I
QoS characteristics in 5G

Resource Type	Packet Delay Budget	Packet Error Rate	Example Services
GBR	100 ms	10^{-2}	Conversational Voice
GBR	50 ms	10^{-3}	Real Time Gaming
GBR	50 ms	10^{-2}	V2X Messages
Non-GBR	300 ms	10^{-6}	Video (Buffered Streaming)
Non-GBR	200 ms	10^{-6}	Mission Critical Data
Non-GBR	100 ms	10^{-6}	IMS Signalling

The user issues a USR to MVNO, which is translated into a network-slice request, and then the MVNO asks Infrastructure Provider (InP) for the resources needed by the slice. Admitting more number of users without requesting extra resources from the InP is crucial for the MVNO, as it maximizes the profits earned. To maximize the benefits, the MVNO can prioritize the guaranteed service users (ex: GBR) over best-effort users (ex: NGBR). MVNO needs to estimate the right amount of resources required to serve each user. The pattern of user arrivals and their diversity collected over long periods can help determine the necessary amount of resources needed. This estimation can be done efficiently by using forecasting techniques. However, sometimes, due to forecasting errors, the resources can be either over-provisioned or under-provisioned. To combat this situation, we also need an efficient slice resource re-configuration mechanism.

In this work, we propose an MVNO Slice Resource Allocation Architecture (MSRAA) considering the inter-slice resource allocation using resource forecasting techniques. By doing Inter-Slice Admission Control (AC), the criterion for admitting incoming QoS flows is decided for the Service Based Architecture of 5G (SBA-5G). The proposed architecture provides both bandwidth and delay guarantees.

We design and implement the MSRAA in alignment with 5G QoS model as specified in [1] by extending upon our previous work [2] in which we prototype the SBA of 5G Core

using open source tools in NFV environment. We evaluate our proposed architecture by testing on two eMBB network slices. While one eMBB slice deals with the GBR traffic, the other one handles the Non-GBR (NGBR) traffic. We evaluate USR acceptances in the slices and do a techno-economic analysis of the MSRAA. It is to be noted that in the following sections, we use the terms flows and USRs interchangeably.

II. Related Work and Motivation

This section discusses a few works done on flow level QoS (w.r.t. bandwidth and delay) and slice & flow-based admission control schemes proposed in the past.

The authors in [3] model the latency distribution of one VNF in Service Function Chains (SFC) as a random-forest regression model which guarantees end-to-end latency and bandwidth consumption of users.

In [4], the authors propose an SDN-based approach for application-based bandwidth allocation where users can allocate upstream and downstream bandwidths for different applications at a high level, offloading application identification to an SDN controller that dynamically installs traffic shaping rules for application flows.

The authors in [5] propose an architectural solution for inter-slice admission and congestion control that copes with existing pre-standardized 5G architectures. The authors in [6] apply forecasting techniques during the admission control to adjust the allocated slice resources to optimize the network utilization while meeting SLAs of network slices.

The authors in [7] propose a bounding admission control strategy to divert blocked traffic in the eMBB frequency band (FB) to overflow to the URLLC FB, in the RAN. In our work, we extend this idea for GBR & NGBR eMBB slices in the data plane of the 5G core network.

Most of the works, as mentioned above, consider only either bandwidth or delay requirements during the inter-slice admission control, but not both. None of the works talk about the techno-economic benefits of their proposed schemes. Besides, the evaluation of the schemes proposed above is either analytical or simulation-based, which doesn't show the practical feasibility of the scheme. Complex schemes such as optimization models take a long time to compute the admissibility, which is not feasible as quick decision times are preferred.

In this work, we propose an MVNO Slice Resource Allocation Architecture (MSRAA) for SBA-5G, which guarantees both the bandwidth and delay requirements of each accepted user flow. Using machine-learning, we predict the future bandwidth requirements and the current E2E delay and use it in our admission control algorithm to maximize acceptance of incoming QoS flows and thus reduce the resource consumption of the slice.

Our main contributions are summarized as follows:-

- Proposing an MSRAA for SBA-5G based on bandwidth and delay guarantees.
- Proposing an architecture for SLA based admission control for USRs in network slices.
- Testbed based evaluation of the proposed scheme in the REST-based SBA-5G.

III. Background

This section discusses the 5G QoS model as defined by 3GPP in [1], and flow level admission control for a network slice in terms of bandwidth and delay requirements.

A. The 5G QoS Model

The standard functionality of Session Management, i.e., assigning an IP address to the user pretty much still works the same as in LTE and 5G. In LTE, the operation was referred to as establishing a Default Bearer. In the 5G core, this operation is now referred to as establishing a PDU Session (PDU = Protocol Data Unit). 3GPP has described the 5G QoS model in [1]. QoS flow is the finest granularity of QoS differentiation in the PDU session. A QoS Flow ID (QFI) is used to distinguish a QoS flow in the 5G system. User Plane traffic with the same QFI within a PDU session gets the identical traffic forwarding treatment. The QFI is provided in an encapsulation header on N3 interface (shown in Fig. 8), i.e., without any modifications to the E2E packet header. Within the 5G system, a QoS flow is controlled by the SMF and may be preconfigured, or set via the PDU Session Establishment procedure. For each GBR QoS flow, the QoS profile includes the following QoS parameters: GBR-Uplink (UL) and Downlink (DL), Maximum Bit Rate (MBR), Delay requirements.

B. Admission Control Framework for SBA-5G

The role of Admission Control (AC) in 5G core network is to analyze the available physical and virtual resources along with their remaining capacity and to decide whether they are capable of accommodating an incoming flow request. AC happens during the PDU session creation. In this work, we are considering elastic flows (flows that end), i.e., we know how long the duration of each flow is. We assume that a QoS flow comes with bandwidth and E2E delay requirements. Therefore, we model the input vector for each flow as (GBR, MBR, flow-duration, delay-requirements).

Measurement Based Admission Control (MBAC): MBAC methods [8] use metrics of the current state of network traffic to support incoming data flows. MBAC methods use the metrics of the traffic and QoS parameters to make decisions for AC. Bandwidth based admission control (BAC) is decided based on the flow's GBR, total bandwidth being used by the existing flows, and total available link bandwidth. Eqns (1) and (2) are used to determine whether to accept or reject the flow, in terms of bandwidth.

$$(C_{measured} + GBR_{newflow}) \times \alpha < C_{total} \quad (1)$$

$$MBR_{newflow} \leq C_{total} \quad (2)$$

where:

α is Admission Policy Factor ($\alpha > 1$),
C_{total} is total available link bandwidth,

$C_{measured}$ is total bandwidth measured from the existing flows.

The admission policy factor α assists in improving the admission control decisions. Using this factor, it can ensured that the total bandwidth used after admitting the flow is always less than C_{total}.

In delay-based Admission Control (DAC), we accept a request if we observe that a particular instance of network slice can satisfy the delay requirements of the flow.

IV. Proposed Architecture

In this section, we propose MVNO Slice Resource Allocation Architecture (MSRAA), to efficiently allocate the slice resources and satisfy the SLA requirements of the flows concerning bandwidth and delay. As shown in Fig. 1, we design three fundamental building blocks: (i) a forecasting module that predicts the future network slices bandwidth requirements based on past traffic, (ii) a network slicing admission control algorithm considering the SLAs of USRs, and (iii) a network slicing resource configuration module which helps in reallocating resources from lower priority slices to a higher priority slice. We explain in detail about each in the following subsections.

A. Forecasting Engine (FE)

Time-Series data is an indispensable part of any network architecture. Often predictions need to be made by analyzing time-series data for optimizing resource utilization. According to [9], many classical time series techniques available such as ARIMA, Holts Winter, etc. can be used for predicting time-series data, but the problem with these approaches is that they assume that the data is correlated. These techniques work well for short-term prediction, but do not prove to be effective for long term data. Due to the dynamic nature of today's network conditions, it is better to go with Deep Learning Techniques to model the non-linear co-relation between the past and the current data points. Hence, in our FE, we use Long Term Short Memory (LSTM) [10] to predict the bandwidth requirements of future time windows and allocate bandwidth based on the predicted values.

The FE is trained based on the bandwidth utilization of all the previous time windows and forecasts the set of bandwidths needed for the next time window. The prediction of bandwidth utilization is made separately for each slice. After each time window, the FE is re-trained on the previous time window, which helps in improving its accuracy over time.

B. Network Slice Resource Configuration (NSRC)

NSRC dynamically allocates the FE's predicted bandwidth to the respective slice in the network. However, sometimes due to forecasting errors, the bandwidth can be under-provisioned. To combat this situation, we also need an efficient slice resource re-configuration mechanism to reallocate bandwidth from lower priority slices to higher priority slices. For this work, we have prioritized eMBB-GBR slices over eMBB-Non-GBR slices, because eMBB-GBR slices provide more profits to the MVNO.

When the rejection rate in the higher-priority slice crosses a certain threshold, the AC informs the NSRC to reallocate resources. The NSRC then reallocates the resources to the high-priority slice, by reducing resources from the low-priority slice(s). To avoid starvation of lower priority slices, if the rejection rate in the lower-priority slice crosses a certain threshold, the NSRC does not proceed with any more reallocations. These re-configured bandwidth details of each slice are informed to the AC. The NSRC also monitors the actual bandwidth being utilized in a time-window and conveys it to the FE for re-training the LSTM model.

C. Admission Control (AC)

The AC is performed for each USR based on the SLA requirements of USR. AC happens during the PDU session creation of the USR. Fig. 2 shows our proposed framework for AC. AC for each flow occurs separately during the PDU session creation time when AMF sends the session creation request to SMF. Before creating the PDU session, the SMF asks the AC whether the flow with the required SLA requirements can be admitted or not. We deal with the flow bandwidth AC (BAC) and flow delay AC (DAC) separately. If sufficient resources are not available, the AC rejects the request and informs the SMF, which then does not create the PDU session. If the UE flow can be accepted, the AC returns the slice ID to the SMF, and the SMF assigns an appropriate QFI corresponding to the slice.

1) Bandwidth Admission Control (BAC): We use Algorithm 1 for BAC. The admissibility criterion of bandwidth determines whether or not it is possible to accept a new flow based on the available bandwidth of the link as shown in Eqns. (1) and (2). The new flow would acquire by sharing capacity fairly with the flows already in progress. As shown in Algorithm 1, the BAC is different for eMBB-GBR flows and eMBB-NGBR flows. The admission control is applied to all the links of the network slice, and then the decision is made. The BAC for eMBB-GBR is performed based on GBR required by the USR. For eMBB-NGBR, we do the admission control by considering the current number of flows and assigning each flow, a minimum bandwidth of $ngbrFlowMinBw$. If a flow is getting the bandwidth less than $ngbrFlowMinBw$, it is rejected. The thresholds of $gbrFlowRejectionThreshold$ and $ngbrFlowRejectionThreshold$ are used to enable and disable the bandwidth sharing among slices.

2) Delay Admission Control (DAC): As a part of SLA, each flow comes with a specific E2E delay requirement. Similar to the Bandwidth Admission Control, we have Delay Admission Control which checks if the E2E delay requirements can be guaranteed if that gets admitted. Estimating the delay incurred by the flow beforehand helps prevent the SLA violation during the progress of the flow. For calculating the delay incurred by the flow, we propose two different approaches and select the best among them depending upon the situation.

- Calculation of E2E delay using Queuing Model
- Prediction of E2E delay based on Mondrian Random Forests

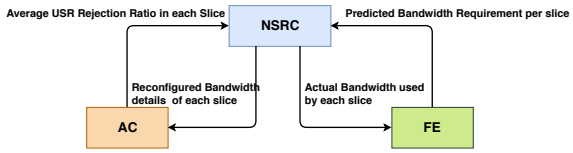

Fig. 1. MVNO Slice Resource Allocation Architecture.

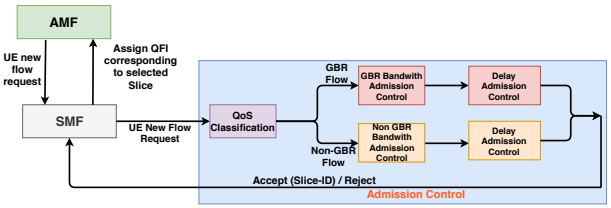

Fig. 2. Admission Control Framework for QoS flows.

Algorithm 1 Bandwidth Admission Control

procedure BAC($request$, $arrival\_time$)
  $admit\_status \leftarrow False$
  if $arrival\_time$ = 0 **then**
    $bw\_used_{GBR} \leftarrow 0$
    $bw\_used_{NGBR} \leftarrow 0$
  if request-type = "GBR" **then**
    if $(request_{GBR}$ + $bw\_used_{GBR}) \times \alpha <$ $total\_bw_{GBR}$ and $request_{MBR} \leq total\_bw_{GBR}$ **then**
      $admit\_status \leftarrow True$
      $currFlowsGbr \leftarrow currFlowsGbr + 1$
      $bw\_used \leftarrow bw\_used + request_{GBR}$
    else
      UpdateRejectionRate()
      if Rejection Rate $\geq$ $gbrFlowRejectionThreshold$ **then**
        Request NSRC to allocate
        bandwidth from eMBB-NGBR slice(s)
  if request-type = "Non-GBR" **then**
    if $total\_bw_{NGBR}$ / $currFlowsNGbr \geq$ $ngbrFlowMinBw$ **then**
      $admit\_status \leftarrow True$
      $currFlowsNGbr \leftarrow currFlowsNGbr + 1$
    else
      UpdateRejectionRate()
      if Rejection Rate $\geq$ $ngbrFlowRejectionThreshold$ **then**
        Request NSRC to stop allocating
        bandwidth from eMBB-NGBR slice(s)
  return $admit\_status$

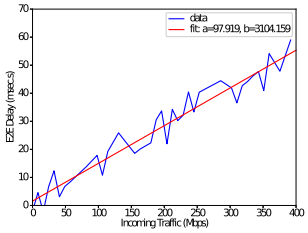

Fig. 3. Curve Fitting of Delays.

Queuing Model for Delay Calculation (QMDC): Here we model our 5GS Core as a queuing system. It is to be noted that queuing happens at the switches placed before UPF and Sink, as shown in Fig. 8. Assuming all the packets arriving in the system follow the Poisson distribution with the single instances of UPF & Sink and maximum bandwidth supported by UPF & Sink as deterministic, we model this system as M/D/1 queuing. In the M/D/1 queuing model, the arrivals are Markovian, the service rate is deterministic, and the number of servers is one. From queuing theory, we define the parameters below:

$$\lambda_{pkt} = TotalIncomingBandwidth/PacketSize \quad (3)$$

$$\mu_{pkt} = MaximumOutgoingBandwidth/PacketSize \quad (4)$$

Here λ_{pkt} is the packet arrival rate and μ_{pkt} is the service rate of UPF and Sink. It is assumed that the packet size is same for all the flows. The packet size and maximum outgoing bandwidth are constants mentioned in Table II.

$$T_{service} = 2 \times (1/\mu_{pkt} + \lambda_{pkt}/2 \times \mu_{pkt}(\mu_{pkt} - \lambda_{pkt})) \quad (5)$$

$T_{service}$ is total time spent by the packet in the network slice. In our setup, a network slice is realized by stitching UPF to RAN interface and UPF to Data Network interface with Open-VSwitch [11] (OvS) bridges (as shown in Fig. 7) . We multiply the total queuing service time($(1/\mu_{pkt}+\lambda_{pkt}/2\times(\mu_{pkt}-\lambda_{pkt}))$ by 2 since the time spent by the packet in the switches of OvS-br1 and OvS-br2 is same (as the OvS flow rules are same in both). From the 5GS implementation framework mentioned in Section-IV, we collected the E2E delay values by varying the traffic load, and we try to fit these values with Eqn. (5). To fit the experimental data we slightly modify it with constants a and b as shown in Eqn. (6).

$$T_{service} = a \times 2 \times (1/\mu_{pkt} + \lambda_{pkt}/2 \times \mu_{pkt}(\mu_{pkt} - \lambda_{pkt})) + b \quad (6)$$

We apply curve-fitting technique using the experimental values and Eqn. (6) and get the values of a and b. We get a=97.9 and b=3104.1 from the curve-fitting as shown in Fig. 3. The values of a and b depend on the distribution of flow bandwidth, duration and arrival pattern which are mentioned in Table II. For evaluation of E2E delay calculation of M/D/1 queuing model, we randomly generate the data from experiments and feed the inputs to Eqn. (6) (with a=97.9 and b=3104.1).

Fig. 4 shows the variation of actual and calculated values of E2E delays. From Fig. 4, it is observed that M/D/1 calculation performs good for small values of traffic, while for large values, we could see many variations. The pattern of the delay is not completely linear, which means that E2E delay is not dependent solely on the incoming traffic.

Mondrian Random Forests based Delay Estimation (MFDE) : We model the E2E delay as an online prediction

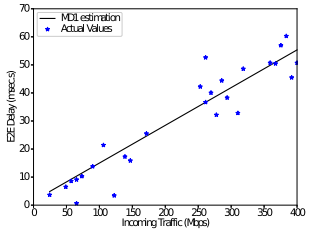

Fig. 4. Delay Calculation using M/D/1 model.

model using Mondrian Random Forest [12]. Mondrian Random Forests are efficient online random forests which achieves competitive predictive performance comparable with existing online random forests and periodically re-trained batch random forests while being more than an order of magnitude faster. We are using Random Forests as our machine learning model because they have low bias and moderate variance, they are robust to outliers, have a quick prediction time and are easily parallelizable. We use the following input parameters for our prediction model.

- CPU utilization of the host machine
- GBR of the new flow
- Number of Flows in the slice

We predict the E2E delay encountered by the flow (from RAN to SINK and back). The DAC algorithm is mentioned in Algorithm 2. After Bandwith Admission Control, the BAC module returns the list of QoS slices which can provide the UE required GBR. The host CPU utilization (cpu_util) and the number of flows in every slice ($num\_flows\_in\_slice[]$) are monitored periodically. For every slice, we predict the E2E delay that the UE flow will experience if it is admitted in the slice. For all the slices we check if the predicted delay is lesser than the required delay and whether the delay requirements of the existing flows in the slice are less than the predicted delay. Out of all the slices which satisfy the above requirements, we select the slice giving the maximum remaining delay. If none of the slices can satisfy our delay requirements, we *reject* the request, else we *accept* the request and return the selected slice to the SMF.

Algorithm 2 Mondrian Random Forest Based Delay Admission Control

```
1:  procedure PRED_DAC(slice_id, req_delay, req_bw)
2:      global Mondrian_Forest
3:      global num_flows_in_slice[]
4:      global cpu_util
5:      X ← [cpu_util, req_bw,
6:              num_flows_in_slice[slice_id] + 1]
7:      pred_delay ← Mondrian_Forest.predict(X)
8:      if pred_delay < req_delay and
9:          checkExistingFlowsDelay(pred_delay) then
10:         return Accept
11:     return Reject
```

During the data transfer, the network conditions and the actual delay that the UE flow is experiencing is monitored by sending periodic cURL requests. After the data transfer has completed, the average delay and average host CPU utilization during the data transfer are calculated and trained in an online manner on the Mondrian Random Forest. We are using an online algorithm because we want our prediction model to learn with time and be robust towards network uncertainties.

Fig. 5 shows the prediction of delay values using MFDE. Fig. 6 shows that as the traffic increases the error percentage is increasing in QMDC but decreasing in MFDE. Since we are using online model, the error get reduced as the model gets more robust with more data.

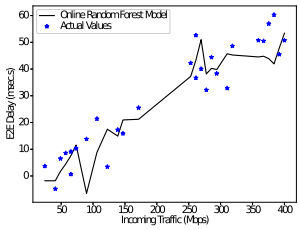

Fig. 5. Delay Calculation using Mondrian Random Forests.

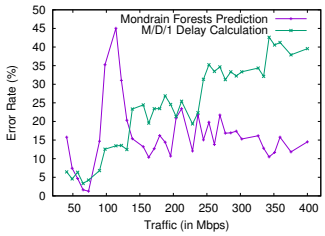

Fig. 6. Error Percentage in QMDC and MFDE.

V. Implementation Framework

This section discusses our implementation framework, which includes the REST-based SBA-5G, realization of the network slices, QoS provisioning of the flows, and MSRAA.

A. REST based SBA-5G

Fig. 8 shows how our REST-based 5G Core architecture (REST-5G-Core) is implemented. The virtualization of resources in the architecture is done with the help of Docker platform.

The REST-5G-Core [13] is designed and implemented as per the standard 3GPP protocol stacks for AMF, AUSF, SMF, UPF, including a RAN simulator, an NRF, and a Sink node. All the NFs are built as software modules in C++11 and run on individual Docker containers on a private cloud. We implement a distributed setup for the Network Function Repository Function (NRF) for service registry and service discovery, using Consul [14], an open-source distributed and highly available service discovery and configuration system. The purpose of Radio Access Network (RAN) simulator is to generate uplink and downlink traffic to the 5GC. It produces multiple threads that simulate UEs, which lead to control plane and data plane transmissions within the 5GC. The Sink node

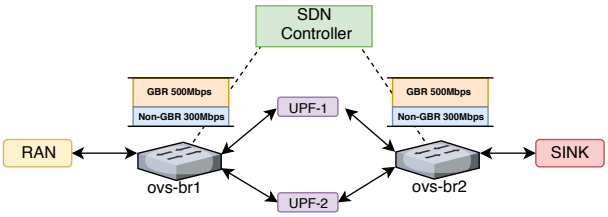

Fig. 7. Fine Grained Bandwidth Allocation.

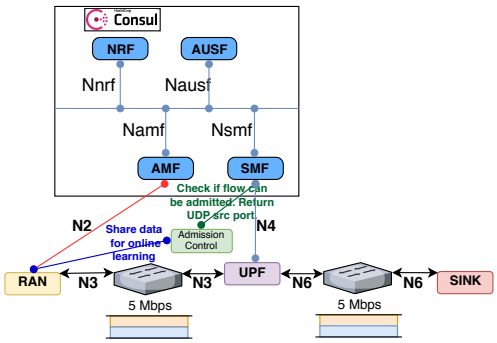

Fig. 8. Implementation Framework.

module is used to serve as a Packet Data Network (PDN) server to receive the generated uplink traffic and send back the acknowledgments as downlink traffic. The modules of AMF, AUSF, and SMF are run on multi-threaded servers to service the requests from other 5GC components and send the responses back. The AUSF simulates the behavior of HSS in LTE-EPC and uses a MySQL database for storing the details of various users.

In this prototype, we have two different types of interfaces.

- *Reference point based interface*: N2, N3, N4, and N6 in Fig. 8 are the reference point based interfaces. Interfaces N2 and N4 follow the multi-threaded Stream Control Transmission Protocol (SCTP) architecture. Interfaces N3 and N6 follow multi-threaded UDP architecture and forwards uplink and downlink traffic between the RAN Simulator Sink. An OpenVPN [15] tunnel is set up between the RAN Simulator and Sink, and the packets flowing via N3 and N6 are encapsulated into UDP packets.
- *Service Based Interface*: Service-based interfaces follow the multi-threaded REST architecture. NAmf, NSmf, NAusf, and NNrf in Fig. 8 are the service-based interfaces. Any module providing any service starts a REST server (operating on HTTP/2) and exposes different API endpoints for the different operations involved in 5G bearer setup. Every service consumer accesses the exposed HTTP endpoint using a cURL request.

When a new UE arrives, the 5GC control plane authenticates the UE and creates a session. At the end of session creation, UEs are ready to do data transfer. An OpenVPN tunnel is set up between the RAN Simulator & Sink, and the data flow is generated synthetically using iperf3 which pumps TCP traffic from the RAN simulator. Each UE traffic flows via a network slice in the 5G data plane.

B. QoS Provisioning of Flows

1) Realization of Network Slice: Each network slice consists of a UPF Docker container which is connected to the RAN simulator and the Sink Docker container by Open-VSwitch [11] (OvS) bridges. Each OvS bridge has OvS QoS slices for eMBB-GBR and eMBB-NGBR slices. Each QoS slice has a Maximum Bit Rate (MBR) specified at the slice creation time, and this MBR is divided equally among all the different UE traffic flows flowing through the slice.

2) Bandwidth Provisioning of flows: Since the N3 and N6 interfaces encapsulate UE traffic onto UDP packets, we use the UDP source port to differentiate UE traffic flows in the OvS flow rules. Each slice is assigned a UDP port range, and we add flow rules in the OvS switch which reads the UDP source port and direct the UE packet onto the corresponding slice.

During admission control of a UE flow, the slice in which the UE flow should be allocated is determined, and then the SMF assigns a corresponding UDP port in the particular slice's port range. The UDP source port is the QFI for that UE flow. The UE traffic is then encapsulated onto UDP packets with the assigned source port. We use the same technique to direct downlink traffic too.

3) Delay Provisioning of Flows : The Admission Control module runs Mondrian Random Forests [12] for Delay-Aware Admission Control.

When a new UE wants to get admitted into the system, the E2E delay that it will face based on the current network slice conditions is predicted. Only those network slices are selected which respect the delay constraints of the UE flow, and the slice is chosen, which provides the maximum remaining delay. During the data transfer, the network conditions and the actual delay that the UE flow is experiencing is monitored by sending periodic cURL requests from RAN to Sink.

After the data transfer has completed, the average delay is taken and trained in an online manner on the Mondrian Random Forests. By doing this, the prediction algorithm becomes more robust to network uncertainties.

C. Realization of FE

The FE forecasts the total incoming bandwidth requirement for each slice in the next time window. It is implemented in Python with LSTM as forecasting technique. The forecasting is applied individually for eMBB-GBR and eMBB-NGBR slices. The total incoming traffic in a slice is measured with the periodicity of one second for each time window at the OvS-br1 and OvS-br2 shown in Fig. 7.

D. Realization of NSRC

The NSRC is implemented in Python, and it sets the Maximum Bit Rate (MBR) in a slice, which is achieved

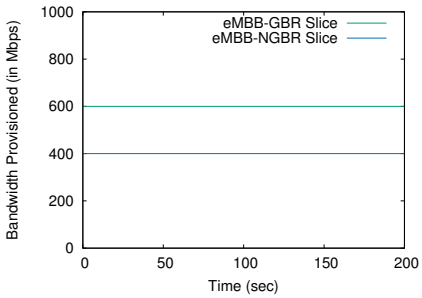

Fig. 9. Bandwidth Provisioned in Scheme-1.

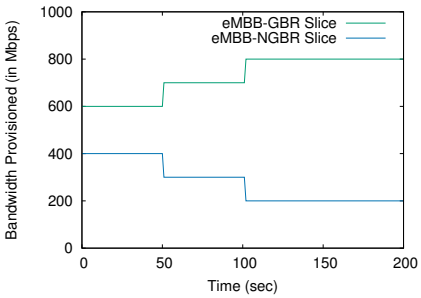

Fig. 10. Bandwidth Provisioned in Scheme-2.

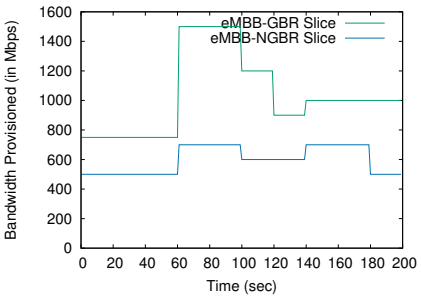

Fig. 11. Bandwidth Provisioned in Scheme-3.

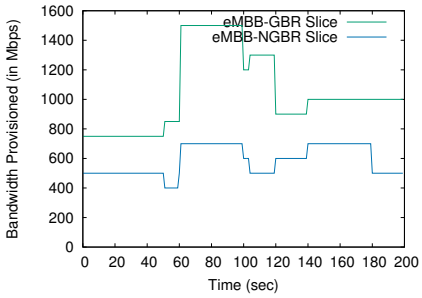

Fig. 12. Bandwidth Provisioned in Scheme-4.

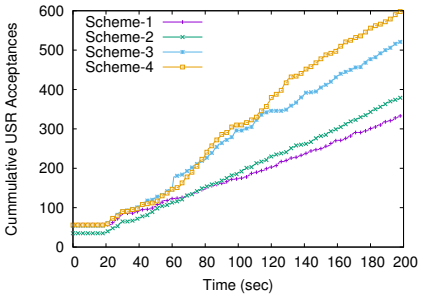

Fig. 13. USR acceptances in eMBB-GBR slice.

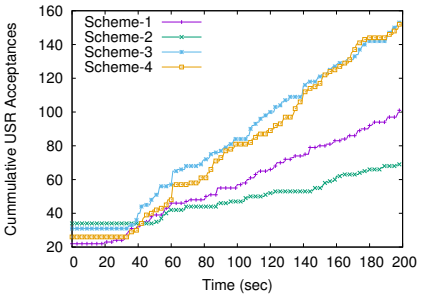

Fig. 14. USR acceptances in eMBB-NGBR slice.

through configuration of MBR of OvS slices in OvS-br1 and OvS-br2 with the help of *ovs-vsctl* Linux commands. If the AC rejection threshold of eMBB-GBR is reached, then the MBR value of eMBB-GBR is increased, and eMBB-NGBR is decreased. After the forecasting is done, the MBR values of eMBB-GBR and eMBB-NGBR are updated based on the forecasted result.

VI. Performance Evaluation

To show the benefit of forecasting and resource reallocation of network slices using MSRAA architecture, we evaluate Algorithms 1 and 2 using Implementation Framework shown in Fig. 8. We denote the various schemes of bandwidth

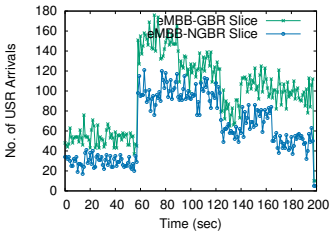

Fig. 15. Arrivals of USRs in slices.

provisioning to the slices by numbering them as follows:

- *Scheme-1* as bandwidth provisioning without reallocation and without forecasting.
- *Scheme-2* as bandwidth provisioning with reallocation and without forecasting.
- *Scheme-3* as bandwidth provisioning without reallocation and with forecasting.
- *Scheme-4* as bandwidth provisioning with reallocation and with forecasting.

The authors in [16] have modeled the daily incoming user traffic pattern at the base station of cellular networks. We emulate the USR traffic in the eMBB-GBR and eMBB-NGBR slices according to that pattern using the Poisson distribution, as shown in Fig. 15.

TABLE II
Experimental Parameters

Parameter	Value
Number of UEs	0 to 300
Simulation time	360 seconds
UE Arrival Distribution	Poisson Distribution
UE data transfer duration	random distribution between [40s,60s]
Virtualization platform	Docker
Traffic generation tool	iperf3
Live status monitor	Prometheous 2.5.0
Packet Size	800 Bytes
$[X,Y,Z]$	[1,0.8,0.1]

We evaluate the Schemes 1 to 4 concerning a number of USR acceptances in eMBB-GBR and eMBB-NGBR slices. We then do the cost-benefit analysis of Schemes 1 to 4.

A. Bandwidth Provisioning and USR Acceptances

- In *Scheme-1*, NSRC allocates fixed values to eMBB-GBR and eMBB-NGBR slices as shown in Fig. 9.
- In *Scheme-2*, NSRC initially allocates fixed bandwidth values to eMBB-GBR and eMBB-NGBR slices. NSRC prioritizes eMBB-GBR slice over eMBB-NGBR slice and reallocates the bandwidth to the slices at t=50 and t=100 as shown in Fig. 10.
- In *Scheme-3*, the provisioned bandwidth values are high when compared to *Schemes 1 and 2* as NSRC allocates based on traffic forecasting as shown in Fig. 11.
- In *Scheme-4*, NSRC initially allocates bandwidth values to eMBB-GBR and eMBB-NGBR slices based on fore-

casting information. NSRC prioritizes eMBB-GBR slice over eMBB-NGBR slice and reallocates the bandwidth to the slices at t=50 and t=100 as shown in Fig. 12.

1) USR Acceptances in eMBB-GBR: Fig. 13 shows USR acceptances in the eMBB-GBR slice. The number of USR acceptances in the eMBB-GBR slice is increasing with the reallocation of bandwidth of the eMBB-NGBR slice. With the help of LSTM Forecasting, a higher number of USR acceptance is observed, it even increases with the redistribution of bandwidth from the eMBB-NGBR slice.

2) USR Acceptances in eMBB-NGBR: Fig. 14 shows USR acceptances in eMBB-NGBR slice. The number of USR acceptances in the eMBB-NGBR slice is decreasing with the reallocation of bandwidth of the eMBB-NGBR slice to the eMBB-GBR slice. With the help of LSTM Forecasting, a higher number of USR acceptance is observed in the eMBB-NGBR slice.

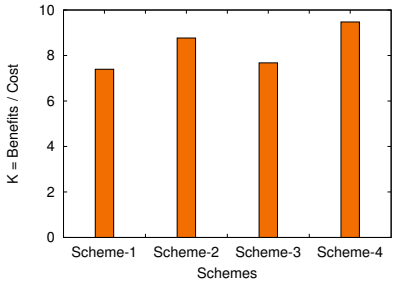

Fig. 16. K values for various schemes.

B. Cost-Benefit Analysis

The cost of MVNO to InP is calculated from the bandwidth used by the MVNO over a period of time. Assuming Z is the cost per bandwidth used, the total cost is found with the help of Eqn. (7).

$$CostToMVNO = Z \times TotalTimeBandwidthUsed \quad (7)$$

In the eMBB-GBR slice, the user pays according to the GBR. Assuming X as a benefit from per GBR flow.

$$BenefitFromGBRFlow = X \times TotalDataUsed \quad (8)$$

The user pays according to the data used in the eMBB-NGBR slice. Assuming Y as benefit per Non-GBR flow then

$$BenefitFromNonGBRFlow = Y \times TotalDataUsed \quad (9)$$

It is safe to assume that $Z < Y < X$ since the MVNO wants to pay less than the benefit from the USR. The values of X, Y, and Z are mentioned in Table II. We pump the traffic into 5G System and measure the total cost and benefit using X, Y, Z values for the total time. As we are trying to maximize the benefits and minimize the costs of the MVNO, we consider K as the total-benefits to total-cost ratio and try to maximize this ratio. Fig. 16 shows the K values for various schemes. From Fig. 16, we conclude that the inter slice resource reallocation mechanism accompanied by the forecasting can increase the benefits and also reduce the cost.

VII. Conclusions

In this work, we proposed the MSRAA by considering inter-slice resource allocation and resource forecasting techniques for the SBA-5G. The application of ML techniques to forecast next time windows and delay prediction is applied in our proposed AC framework. It is shown that for predicting delay of the flow, the Mondrain Random Forests perform better than traditional queuing models (as shown in Fig. 6). By considering two network slices of eMBB-GBR and eMBB-NGBR, we do reallocation of bandwidth resources among the eMBB-GBR and eMBB-NGBR network slices to increase the acceptance of GBR flows, thus leading to more profits with limited starvation of Non-GBR flows. We show that reallocation of resources among eMBB slices accompanied by the ML techniques increases profit and reduces costs (from Fig. 16).

References

[1] 3GPP, "3GPP TS 23.501 - System Architecture for the 5G System," 2018.

[2] T. V. Kiran Buyakar, H. Agarwal, B. R. Tamma, and A. F. A, "Prototyping and load balancing the service based architecture of 5g core using nfv," in *IEEE Conference on Network Softwarization (NetSoft)*, June 2019, pp. 228–232.

[3] T.-H. Lei, Y.-T. Hsu, I.-C. Wang, and C. H.-P. Wen, "Deploying qos-assured service function chains with stochastic prediction models on vnf latency," in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2017, pp. 1–6.

[4] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, and Y.-Q. Song, "Flowqos: Per-flow quality of service for broadband access networks," Georgia Institute of Technology, Tech. Rep., 2015.

[5] B. Han, A. DeDomenico, G. Dandachi, A. Drosou, D. Tzovaras, R. Querio, F. Moggio, O. Bulakci, and H. D. Schotten, "Admission and congestion control for 5g network slicing," in *IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2018, pp. 1–6.

[6] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5g network slicing resource utilization," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[7] Yuguang Fang and Yi Zhang, "Call admission control schemes and performance analysis in wireless mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 371–382, March 2002.

[8] E. Chromy, M. Jadron, and T. Behul, "Admission control methods in ip networks," *Advances in Multimedia*, vol. 2013, p. 2, 2013.

[9] "Comparison between classical statistical model (arima) and deep learning techniques (rnn, lstm) for time series forecasting." [Online]. Available: https://www.linkedin.com/pulse/comparison-between-classical-statistical-model-arima-deep-virmani/

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[11] L. Foundation, "OpenVSwitch," https://www.openvswitch.org/, 2019.

[12] B. Lakshminarayanan, D. M. Roy, and Y. W. Teh, "Mondrian forests: Efficient online random forests," 2014.

[13] "5g-core-rest-sba," https://github.com/sipian/5G-Core-Rest-SBA.

[14] HashiCorp, "Consul," https://consul.io/, 2018.

[15] M. Feilner, *OpenVPN: Building and integrating virtual private networks*. Packt Publishing Ltd, 2006.

[16] S. Morosi, P. Piunti, and E. Del Re, "Improving cellular network energy efficiency by joint management of sleep mode and transmission power," in *24th Tyrrhenian International Workshop on Digital Communications-Green ICT (TIWDC)*. IEEE, 2013, pp. 1–6.