A Semi-Empirical Data-Rate Estimation Method of 5G RAN Slicing

Wen-Ping Lai[*,#], Ming-Jay Lai[+], and Hong-Lun Lai[*]
[*]Department of Electrical Engineering, Yuan Ze University, Taoyuan City, Taiwan
[+]Department of Communication Engineering, National Central University, Taoyuan City, Taiwan
[#]Email: wpl@saturn.yzu.edu.tw

***Abstract*—Network Slicing is a novel 5G technique enabling a single net to meet differential needs from multiple tenants. In this paper, a *semi-empirical* method called *linear reduction factor* (LRF) for the estimation and prediction of data goodput rates is proposed for *radio access network* (RAN) slicing. The method is derived from our fundamental understanding of peak data throughput calculation, and a bold but ambitious assumption that the *linearity* of the *RAN-sliced data goodput* should hold with the amount of allocated resource, in units of *resource block group* (RBG). The paper presents theoretical derivation steps in a semi-empirical way toward how to estimate the *slice goodput* directly from the *measured reduction factor* of *system throughput* with the full frequency bandwidth. When combined the *linearity* with *the number of allocated RBGs*, the proposed method can precisely predict the *data goodput for a RAN slice* with any given number of RBGs. This research also successfully ran a *software-defined* platform to conduct the verification experiments using the Mosaic5G FlexRAN real-time controller for configuring and monitoring the emulated RAN. The desired *linearity* and *precision* of the proposed LRF method are fully supported by two case studies, namely *equal-slicing* and *prioritized-slicing*.**

***Keywords*—5G, SDN, NFV, RAN slicing, LRF**

I. Introduction

The new era of the *fifth generation* (5G) communication technology [1, 2] is remarkable for its flexibly adaptive capability to meet diversified and differential service needs, coming not only from individual users but also from enterprise tenants, such as *over-the-top* (OTT) service providers or stakeholders for *industrial internet of things* (IIoT). Such a remarkable capability is the major enabler of paradigm shift to 5G digital transformation, where a flexible and customizable communication system is possible and achievable.

The 5G communication system should be in essence equipped with some level of programmable control to transform a single physical net into multiple virtual nets, which can thus be independently and logically owned by multi-tenants of differential needs, but actually share the same pool of physical resources, such as computing, memory, networking and radio. In other words, the above physical resources should be appropriately *sliced* in an on-demand manner among multiple tenants, together with some level of administrative authorization open to the tenant owners so that they can run their own business in the sense of *mobile virtual network operators* (MVNOs). From the perspective of telecom operators, in particular, the radio resource is the most expensive one and thus the most precious one. Hence, how to efficiently allocate the precious radio bandwidth to multi-tenants forms the major issue of *radio access network* (RAN) *slicing*.

To tackle the issue of RAN slicing, novel networking technologies should be adopted, such as *software defined networking* (SDN) and *network functions virtualization* (NFV) [3-5]. SDN focuses on the *separation* of *control plane* and *data plane* (aka *user plane* in terms of telecom) in order to achieve a central, remote and programmable controller over many distributed switching and routing devices. As a counter partner of SDN, NFV emphasizes the decoupling of software and hardware so as to transform the conventional scheme of closed software bound to proprietary hardware into a novel scheme of open-source software and open hardware, aiming at lower down *capital expenditure* (CAPEX) and *operating expenditure* (OPEX) while enhancing high operation efficiency or even *zero-maintenance agility* in response to versatile and differential user demands. More specifically speaking, NFV releases the *physical network functions* (PNFs) from the vendor-lock-in proprietary hardware using *virtual network functions* (VNFs) so as to achieve *modularization*, *softwarization* and *virtualization* for agile deployment and flexible operation of network functions and services [6]. In general, SDN was initially developed for next-generation datacom, but also triggered the formation of an international operator forum (i.e., the ETSI NFV) toward telecom openness for the future. Nowadays both SDN and NFV have been widely perceived as promising solutions for the 5G digital transformation in all aspects, including the *core network* (CN) [7], the *mobile edge cloudlet* (MEC) [8-10], and even the RAN, where *network slicing* [11-13] has become a common technical language both academically and industrially. Not surprisingly at all, they all come with different technical details too. This paper will focus on the issue of RAN slicing.

CellVisor [14], SoftRAN [15] and RadioVisor [16] are three early works toward *software-defined* (SDx) *telecom networking*, SDx *virtual base stations* and *RAN slicing* respectively. CellVisor pointed out that the major drawback of 4G networking lies in the fact that the user plane is *over centralized*, and the control plane is *over distributed*. It thus proposed a *centralized controller* for the control plane, and introduced a concept called *local agents* on all the distributed *base stations* (eNodeBs) and *serving gateways* (S-GWs) to offload the merging traffic going into a single *PDN gateway* (P-GW), and thus shorten the latency between the RAN and the CN, which is similar to the concept of MEC today. However, it did not address the issue of RAN slicing. SoftRAN proposed a concept called *virtual macro cell,* which represents the central brain of all the small cells deployed within an area, and introduced a SDN control plane to collect RAN information base (RIB), including the interference map, traffic recording, user information and operator preferred polices so as to manage the allocation of *resource grids* formed by (*time*, *frequency*, *cell ID*), whereas individual small cells are simplified to form the user plane with only a minimum control logic. Rooted from SoftRAN, RadioVisor focused on RAN slicing and virtualization, and emphasized that each MVNO should be able to run their own RAN slice so that multi-tenant scheduling should be considered to guarantee independent management and fair resource sharing. In spite of novelties, all the above works were simply conceptual, and no realization was provided.

FlexRAN [17], a work item proposed by the Eurecom Mosaic5G project [18-19], realized the aforementioned concept of *virtual macro cell* where the collection and applications of RIB have been designed and implemented in detail, and provided a good SDN software framework for pursuing coordination among multiple cells. Based on the Eurecom *open air interface* (OAI) project [20], contributing open-source based 4G/5G software for running with generic x86 PC or server hardware, FlexRAN was realized on top of the joint platform of OAI-RAN and OAI-CN. FlexRAN can in essence provide a flexibly SDx RAN which has conquered three realization challenges: (1) clear separation of the mixed control and user planes of OAI-RAN, (2) both channel-condition and network-condition aware adaptation of RAN control functions for dynamic configuration and planning, and (3) highly real-time deployment of network functions in terms of two-layer central and agent controllers, where the former is the so-called FlexRAN central controller, and the latter is the so-called local agent, living together with individual cells. The next move of FlexRAN is toward FlexVRAN [21], further targeting at the deployment of *virtual macro cell* in a heterogeneous networking environment.

Network Store [22], another work item of Mosaic5G, complements the NFV software frame work serving as an on-line APP-like market place of *virtual network functions* (VNFs), which can be requested and downloaded on demand via the RESTful API of FlexRAN controller, and then passed onto the local agent of the base station, and eventually increase a new component function of the base station. In other words, it is possible to modify a component function of the base station in a manner of hot plug-and-play.

This paper proposes a semi-empirical (S.E.) method called *linear reduction factor* (LRF) for the prediction of data goodput rate in the context of RAN slicing. The method is derived from our fundamental understanding of peak data throughput calculation and a bold but ambitious assumption that the *linearity* of the *sliced data goodput* should hold with the amount of allocated resource, in units of *resource block group* (RBG). This paper also presented two case studies, namely *equal-slicing* and *prioritized-slicing*, where the desired linearity and prediction precision of the proposed method are fully supported by the measurement results.

The remainder of this paper is organized as follows. Section II derives the proposed method. Section III describes the experimental environment, results and analyses. Section IV concludes this paper and outlooks into the future.

II. Proposed Method (LRF) for RAN-Sliced Goodput

The proposed LRF method has two novel key concepts: (1) the *linearity* of slice goodput with the number of RBs (or RBGs more specifically), and (2) the *semi-empirical* replacement of the *calculation*-based goodput per RB by the *measurement*-based goodput per RB. Concept (1) claims that the *linearity* of *system goodput* also applies to the *linearity* of *slice goodput*, which forms the theoretical derivation basis of Eqs. (6) and (7) from Eqs. (2) and (3). Concept (2) claims that there exists a measurable *reduction factor* (F_R), which is the ratio of the actual system goodput to the ideal system throughput, as defined by Eq. (5). The following subsections explain why and how LRF has been developed.

A. Complicated Factors of System Goodput

For the conventional case where RAN is not sliced, all the users share the entire frequency resource of the base station, where a *physical resource block* (PRB) is the smallest allocation unit. In principle, the throughput of a PRB can be affected by factors such as *bandwidth*, *channel quality*, *network load*, etc. In fact, a PRB is a rectangle in the frequency-time domain, which spans over 12 OFDM subcarriers in frequency (i.e., 180 kHz = 12 subcarriers $\times$15 kHz/subcarrier) and 7 symbols in a time slot of 0.5 ms for the normal (long) *cyclic prefix* (CP) in a rural area, where CP helps the receiver better correlate the received symbols and overcome the self-interference. In other words, taking 4G as a familiar example, a PRB consists of 84 (= 12 $\times$ 7) resource elements (REs), each of which transmits 2, 4 or 6 bits (aka *modulation order*, denoted as Qm) with the modulation scheme being QPSK, 16 QAM or 64 QAM, respectively for the lowest to highest channel qualities in between each user equipment (UE) and its associated base station.

From the perspective of scheduling, the smallest scheduling unit is a *scheduling resource block* (SRB, or RB for short), equivalent to the size of two PRBs in the time domain. RBs can be allocated to the UE during each *transmission time interval* (TTI) (e.g., fixed to 1 ms for 4G, and variable for 5G). Given the aforementioned information, it is thus easy to carry out by calculation that the peak data rate of one RB is around 1 Mbps. However, such a peak data rate is not practical yet since it only considers the perfect channel condition.

As shown by Eq. (1), other degradation factors such as the REs reserved for the corresponding Code Rate for forward error correction at error-prone channel conditions, and the control-plane and data-plane protocol overheads etc., will further reduce the data throughput achievable by one RB. It is thus complicated to calculate the *throughput per RB* or even *goodput per RB* since it involves with considering these *degradation factors* as multi-layer implicit functional relations of the time-varying channel condition, actually implemented with many lookup tables by 3GPP TS 38.214 [23].

$$\mathrm{R_{th}} = \mathrm{R_{th}}\left(\mathrm{n_{Sched.RBs}}, \mathrm{R_{RB}}\left(\mathrm{N_{OH}}, \mathrm{I_{MCS}}\left(\mathrm{I_{CQI}}(\mathrm{t})\right)\right)\right) \quad (1)$$

Table I: Notation-and-Meaning Table for Eq. (1).

Notation	Meaning
$\mathrm{R_{th}}$	Goodput per UE
$\mathrm{n_{Sched.RBs}}$	Number of Scheduled RBs
$\mathrm{R_{RB}}$	Goodput per RB
$\mathrm{N_{OH}}$	Number of bits for Control-plane and Data-plane Overheads
$\mathrm{I_{MCS}}$	Index of Modulation and Coding Scheme
$\mathrm{I_{CQI}(t)}$	Index of Channel Quality Indicator (running within 1 ~ 15)

B. *Simplification of Goodput by Linearity and Semi-Empirical Replacement*

Given the complications of Eq. (1), two types of simplification are thus strongly needed as shown in Eqs. (2) and (3).

- Linearity described by $\mathrm{n_{Sched.RBs}}$
  As shown by Eq. (2), we assume that the term of $\mathrm{n_{Sched.RBs}}$ can be *factored out* from the $\mathrm{R_{th}}$ function in Eq. (1), based on a simple intuition that $\mathrm{R_{th}}$ should be *linearly* proportional to $\mathrm{n_{Sched.RBs}}$. Meanwhile, $\mathrm{n_{Sched.RBs}}$ equals the product of $\mathrm{N_{RBs}}(\Delta\mathrm{f})$ and $\mathrm{P_{Sched.}}$. Note that $\mathrm{n_{Sched.RBs}} = \mathrm{N_{RBs}}(\Delta\mathrm{f})$ in the context of single UE where $\mathrm{P_{Sched.}} = 100\%$.
- Semi-empirical replacement of $\mathrm{R_{RB}}$ by $\mathrm{R_{RB}^{S.E.}}$
  As shown in Eq. (3), to skip all the complications in the multi-layer functional relations and determine the goodput per RB without knowing all the great technical details, we propose to replace the *calculation*-based $\mathrm{R_{RB}}$ by the *measurement*-based $\mathrm{R_{RB}^{S.E.}}$, where the notation S.E. in the superscript means that such a quantity is *semi-empirical*, i.e., obtained directly from the *measurement* of *system goodput*.

$$\mathrm{R_{th}} = \mathrm{n_{Sched.RBs}} \cdot \mathrm{R_{RB}} \quad (2)$$
$$= \mathrm{N_{RBs}}(\Delta\mathrm{f}) \cdot \mathrm{P_{Sched.}} \cdot \mathrm{R_{RB}^{S.E.}} \quad (3)$$

Table II: Notation-and-Meaning Table for Eqs. (2) and (3).

Notation	Meaning
$\mathrm{N_{RBs}}(\Delta\mathrm{f})$	Total number of RBs, varying with the system bandwidth Δf
$\mathrm{P_{Sched.}}$	Probability of scheduling for a RB

C. *How to conduct Semi-Empirical Replacement of R_{RB}*

The general form of the $\mathrm{R_{RB}^{S.E.}}$ term of Eq. (3) can be expressed by Eqs. (4) and (5) in the sense that it is the *ideal throughput per RB* (denoted as $\mathrm{R_{RB}^{Ideal}}$) times a *reduction factor* $\mathrm{F_R}$, which is a ratio of the *measured* value of *system goodput* (denoted as $\mathrm{R_{th}^{Meas}}$) to the *ideal system throughput* (denoted as $\mathrm{R_{th}^{Ideal}}$) in the context of perfect channel condition identified by $\mathrm{I_{CQI}^{max}}$. Note that $\mathrm{R_{th}^{Meas}}(\mathrm{I_{CQI}^{max}})$ absorbs all the aforementioned *degradation factors*, including the protocol overheads, whereas $\mathrm{R_{th}^{Ideal}}$ does not contain any of them. $\mathrm{F_R}$ can also be generalized to any time-varying channel condition in general. Note that $\mathrm{R_{th}^{Meas}}(\mathrm{I_{CQI}^{max}})$ can be easily measured by a network benchmarking toolset such as *iperf* or *iptraf*, where the former can give the TCP and UDP goodputs and the latter can identify both the goodputs and the Ethernet throughput.

$$\mathrm{R_{RB}^{S.E.}} \equiv \mathrm{R_{RB}^{Ideal}} \cdot \mathrm{F_R} \quad (4)$$

$$\mathrm{F_R} \equiv \mathrm{R_{th}^{Meas}}(\mathrm{I_{CQI}^{max}}) \; / \; \mathrm{R_{th}^{Ideal}} \quad (5)$$

where $\mathrm{R_{RB}^{Ideal}} = \mathrm{N_{SC}} \cdot \mathrm{N_{Sym}} \cdot \mathrm{Q_m^{max}} \; / \; \Delta\mathrm{t_{TTI}}$
$\mathrm{R_{th}^{Ideal}} = \mathrm{N_{RBs}}(\Delta\mathrm{f}) \cdot \mathrm{R_{RB}^{Ideal}}$
with $\mathrm{I_{CQI}^{max}} = 15$ and $\mathrm{Q_m^{max}} = 6$ bits (e.g., for 4G)

Table III: Notation-and-Meaning Table for Eq. (5).

Notation	Meaning
$\mathrm{N_{SC}}$	Number of Subcarriers per RB (i.e., 12)
$\mathrm{N_{Sym}}$	Number of Symbols per RB (e.g. 14)
Qm	Modulation Order (2, 4 , 6 bits for 4G, and up to 8 bits for 5G)

D. *Linear and Semi-Empirical Goodput Prediction of RAN Slices*

Given the linearity of $\mathrm{R_{th}}$ with $\mathrm{n_{Sched.RBs}}$, it is another bold but ambitious assumption that such a linearity can also be universally extended to the cases of RAN slicing (denoted as $\mathrm{R_{th}^{SL}}$) as defined by Eqs. (6) and (7). In other words, both the equations are simply the sliced version of Eqs. (2) and (3), with the S.E. replacement of $\mathrm{R_{RB}}$ by $\mathrm{R_{RB}^{S.E.}}$ again, as shown in Eqs. (4) and (5). To support such a linear and S.E. goodput prediction for RAN slices, verification experiments are given in the next section for two typical cases: *equal slicing* and *prioritized slicing*. Note that $\mathrm{N_{RBs}^{SL}}(\Delta\mathrm{f})$ stands for the number of RBs for a given slice.

$$\mathrm{R_{th}^{SL}} = \mathrm{n_{Sched.RBs}^{SL}} \cdot \mathrm{R_{RB}^{S.E.}}\left(\mathrm{I_{CQI}}(\mathrm{t})\right) \quad (6)$$
$$= \mathrm{N_{RBs}^{SL}}(\Delta\mathrm{f}) \cdot \mathrm{P_{Sched.}} \cdot \mathrm{R_{RB}^{S.E.}}\left(\mathrm{I_{CQI}}(\mathrm{t})\right) \quad (7)$$

E. *Resource Block Group* (RBG)

In practice, according to Table 7.1.6.1-1 of 3GPP TS 36.213 [24], another concept called *resource allocation type* specifies the way where the scheduler allocates RBs for each transmission, and there are three types, among which Type 0 is the simplest way. It classifies multiple RBs into one RBG, where the RBG size (i.e., the number of RBs) varies with the *system bandwidth* (Δf), as shown in Table IV. For instance, for a system bandwidth of 5 MHz, there are 25 RBs in total (namely, $\mathrm{N_{RBs}}(5\ \mathrm{MHz}) = 25$), and thus 13 RBGs are grouped (namely, the ceiling of $\mathrm{N_{RBs}}/\mathrm{S_{RBG}}$ for 5MHz) where each group consists of 2 RBs, except the 13th group, consisting of only 1 RB. All the other cases of system bandwidth work in a similar way. To be a more general form, Eq. (8) describes the relation among N_{RBs}, S_{RBG} and N_{RBGs}. In other words, Eq. (8) can be inserted back into Eq. (7) to replace $N^{SL}_{RBs}(\Delta f)$ by $N^{SL}_{RBGs}(\Delta f)$.

Table IV: Number of RBs per RBG as a function of system bandwidth.

Notation: Meaning	Typical Values					
Δf: System Bandwidth (MHz)	1.4	3	5	10	15	20
$N_{RBs}(\Delta f)$: Total number of RBs	6	15	25	50	75	100
$S_{RBG}(\Delta f)$: Number of RBs per RBG	1	2	2	3	4	4
$N_{RBGs}(\Delta f)$: Number of RBGs	6	8	13	17	19	25

$$N_{RBs}(\Delta f) = \begin{cases} S_{RBG} \cdot N_{RBGs} & \text{for } N_{RBs} \bmod S_{RBG} = 0 \\ S_{RBG} \cdot N_{RBGs} - 1 & \text{for } N_{RBs} \bmod S_{RBG} > 0 \end{cases} \quad (8)$$

III. Results and Analyses

A. Experimental Environment

This section describes the common experimental setup for supporting the verification of the proposed S.E. method in predicting the *slice*-based goodput $R^{SL}_{th,S.E.}$ as discussed in Eqs (6) and (7). Note that the superscript SL in $R^{SL}_{th,S.E.}$ stands for any given slice, and the subscript means that the RAN-sliced goodput is *semi-empirical* since the proposed *reduction ratio* F_R in Eq. (5) is involved in the predicting $R^{SL}_{th,S.E.}$. How to measure F_R is exemplified and obtained in Fig. 3 in the case of non-slicing (namely, the whole system bandwidth of 5 MHz was adopted), and further supported both by Figs. 4 and 5, for *equal slicing* and *prioritized slicing* respectively.

Fig. 1 demonstrates a typical experimental setup of the FlexRAN-based platform for a RAN, with one *base station* (BS) and two associated cellular phones (or phones for short), where the OAI-CN, running separately on a physical machine such as a generic x86 machine (a *laptop computer* in our case), is important to support stable association controls and session connections of phones to the Internet, is not shown here for simplicity. As for the RAN part, the BS and phones can both be physical ones, or both be emulated. At the early stage of this research, a physical combination scheme of RAN slicing has been established and verified successfully, however the wireless condition was difficult to control to give the desired precision level of measurements, as expected in general. Hence, a Layer-2 emulation platform called *l2sim* was adopted instead, while the FlexRAN controller (denoted as *flexran-rtc* in Fig. 1) as a real-time SDN controller for the emulated RAN is still a physical one. While running the experiment, the physical OAI-CN should be started and get ready first, and *flexran-rtc* should then be run to ensure that it can monitor and control the BS and the phones to be deployed. A python-based web-browser GUI dashboard tool was adopted to monitor the working conditions of both *flexran-rtc* and *l2sim*, between which a FlexRAN protocol API can be utilized to *configure*, *reconfigure* and *monitor* the *control-plane* and *user-plane* information of RAN (i.e., RIB, as aforementioned) in real time based on the RESTful HTTP end-point applications, described in the FlexRAN northbound API document [25]. As shown, two phones with their CQIs being 15 and 14 are demonstrated: the one with CQI=15 (the perfect condition) can have an obviously higher throughput and thus a higher goodput than the other.

In fact, a *curl*-based *command line interface* (CLI), together with a JSON information filtering tool, called *jq*, was even more practical to control the RIB and monitor traffic statistics to and from the BS and the phones in real time, using the format of JSON files.

A demonstrated example of a JSON-based configuration file is shown in Fig. 2. As seen, the BS can be *None* or *Static sliced*, meaning that the whole-spectral-band RBGs will be

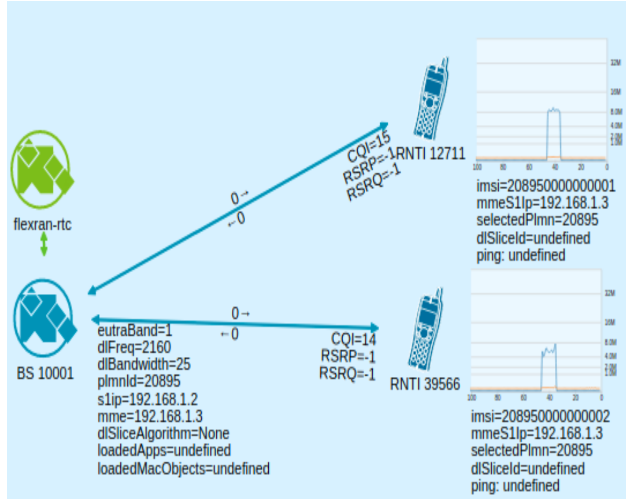

Fig. 1. A typical setup of the FlexRAN-based RAN.

```
{
  "dl": {
    "algorithm": "Static",
    "slices": [
      {
        "id": 0,
        "label": "Slice-1 (8 RBGs)",
        "static": {
          "posLow": 0,
          "posHigh": 7
        }
      },
      {
        "id": 1,
        "label": "Slice-2 (4 RBGs)",
        "static": {
          "posLow": 8,
          "posHigh": 11
        }
      }
    ]
  },
  "ul": {
    "algorithm": "Static",
    "slices": [
      {
        "id": 0,
        "label": "Slice-1 (8 RBGs)",
        "static": {
          "posLow": 0,
          "posHigh": 7
        }
      },
      {
        "id": 1,
        "label": "Slice-2 (4 RBGs)",
        "static": {
          "posLow": 8,
          "posHigh": 11
        }
      }
    ]
  }
}
```

Fig. 2. A typical JSON configuration of *prioritized slicing*.

fully used or just *partly sliced* respectively, where the latter was split into two case studies: *equal slicing* and *prioritized slicing*. As aforementioned, a system bandwidth of 5 MHz was adopted for simplicity, where 25 RBs or 13 RBGs are contained in such a bandwidth. For all the cases of goodput measurement, the downlink TCP goodputs from the PGW of OAI-CN to the UE of *l2sim* were carried out using the *iperf* tool. To be simple and see the linearity of allocated resource, only the *round-robin* scheduler was applied to all the target slices.

B. None-Sliced Throughput and Goodput Measurements

Fig. 3 demonstrates the results of the *None-sliced* case. It is clearly seen that the *measured-TCP-goodput* is around 17.2 Mbps from the aspect of application layer of *iperf*. Note that it was also checked with another tool called *mac_rate* to confirm that the *measured throughput* of the MAC layer is around 18 Mbps, very close to the theoretically calculated value: 18.336 Mbps, obtained from the derived equations plus many lookup tables in the 3GPP TS documents in a complicated way (to consider all the non-perfect channel conditions, MCS schemes, Overhead factors, etc.), as already discussed in Section II.

As shown in Fig. 3, the proposed reduction factor F_R by Eq. (5) was measured and determined to be 68.8% (i.e., 0.688). Based on the reasoning of those equations derived in Section II, we found the S.E measured value of $R_{RB}^{S.E.}$ very powerful when combined with the assumed *linearity* of $n_{Sched.RBs}^{SL}$ (by multiplying the number of allocated RBGs or the equivalent number of RBs) in predicting all the goodputs of RAN slices. The experiment results in the following subsections will further confirm this point.

C. S.E. Prediction of Slice Goodput (*Equal Slicing*)

The previously measured values of F_R (i.e.,0.688) was first tested to *semi-empirically* (S.E.) predict the slice goodput in the case of *equal slicing*, where 2 slices were configured to be the same, i.e., each with 6 RBGs, or the 12 RBs equivalently, in the case of the whole system bandwidth of 5 MHz. In the meantime, each slice was deployed with 2 phones. Based on the term R_{th}^{Ideal} in Eq. (5), the ideal system throughput is expected to be 1 Mbps per RB, and 25 Mbps for 25 RBs. From the perspective of RBGs, each RBG, consisting of 2 RBs, should generate 2Mbps except that the 13th RBG, consisting of only 1 RB, only generates 1 Mbps. In other words, the *ideal throughput* of *each slice* with 6 RBGs should go as high as 12 Mbps. Considering the fact that each slice was shared by the two phones, the *ideal throughput* of *each phone* should go as high as 6 Mbps. As shown by the red dotted lines in Fig. 4, the *ideal slice throughputs* will all be reduced to 4.128 Mbps (i.e., *slice goodputs*) if considering $F_R = 0.688$ according to Eq. (5). Based on the emulation results in green, such an *S.E. prediction* seems to be fully supported by the measurement results if one ignores the inconsistency from the various behaviors of TCP, such as *slow start*, *flow control* and *congestion control*, etc.

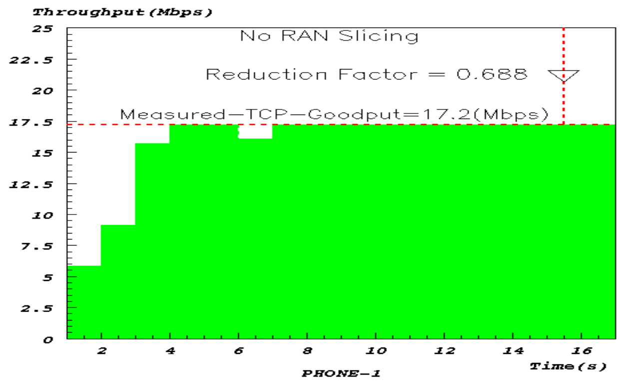

Fig. 3. The *reduction factor* F_R derivable from the Measured-TCP-Goodput in the case of *None Slicing*.

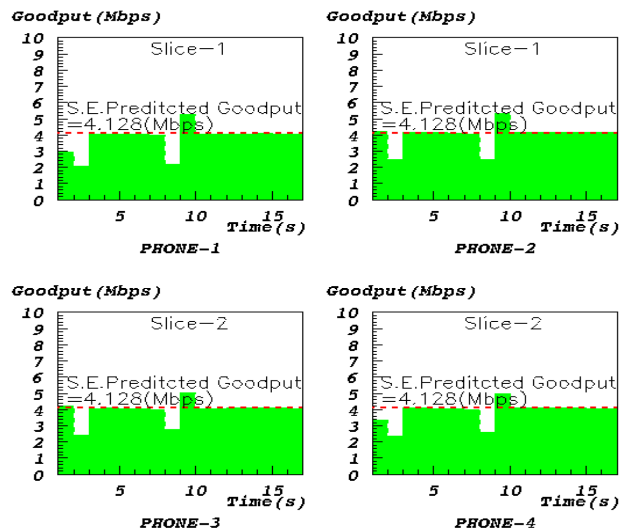

Fig. 4. Confirmation of the *reduction factor* F_R in predicting the Slice Goodput in the case of *Equal Slicing*.

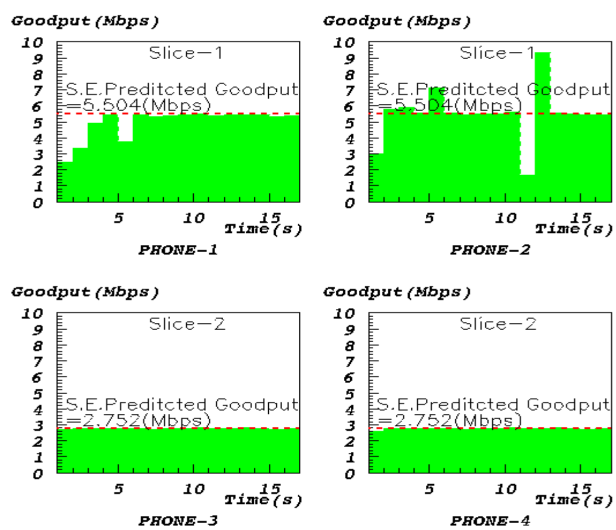

Fig. 5. Confirmation of the *reduction factor* F_R in predicting the Slice Goodput in the case of *Prioritized Slicing*.

D. S.E. Prediction of Slice Goodput (*Prioritized Slicing*)

The amazing journey of such an S.E. prediction was extended to another supporting case: *prioritized slicing*, where two slices were unequally sliced: one was allocated with 8 RGBs, and the other 4 RGBs. Again, based on the nature of *linearity* with the allocated number of RBGs, and thus that of RBs, together with the magic *reduction factor* F_R that we proposed *semi-empirically*, the S.E. predicted values for both the slices seem to be verified again by the measured results, as shown in Fig. 5, where all the green data agree well with the S.E. predicted red dotted lines again.

In general, the *linearity* feature supported by both the test cases of *equal slicing* and *prioritized slicing* implies that generalization to more test cases can also be expected.

IV. Conclusion and Outlook

In this paper, a semi-empirical method called LRF for the prediction of RAN slice goodputs has been proposed. The LRF is built upon two key concepts: (1) the *linearity* of slice goodput with the number of RBGs, and (2) the *semi-empirical* replacement of the *calculation*-based goodput per RB by the *measurement*-based one. The combination of these two concepts has precisely predicted the goodput of a given RAN slice. Two typical cases of RAN slicing, *equal slicing* and *prioritized slicing*, have been successfully verified, where the TCP goodput measurements of both the cases agree well with the semi-empirically predicted values. The success of such a semi-empirical method skips the need of knowing much of tedious technical details, and does provide an intuitive view and a controllable way in facing the era of 5G RAN slicing, coming with a much larger channel bandwidth and thus a much higher data rate.

In particular, the case of *prioritized slicing* is definitely important to the differential service demands for multi-tenant applications. For this objective, more extensive cases of *prioritized slicing* might be needed. Also, tests over more ranges of system bandwidth are also interesting. In addition, it is interesting to know whether the proposed method will also hold for the FR-2 case of 5G NR. Other considerations such as the multiple-access effects within a given slice would also be important. However, this raises up a potential study issue for competition behaviors among users within the same slice, given the good independence among multiple RAN slices. More studies along this track using different schedulers such as *proportional fairness* and *maximum throughput* are underway, where how the fundamental behaviors of these conventional scheduling metrics change within a slice will also need more studies and verifications before the multi-tenant application scenarios of differential needs for the 5G era to be really applicable.

Acknowledgement

This work was supported by Taiwan's Ministry of Science and Technology under grants 108-2221-E-155-022-MY2 and 110-2221-E-155-003.

References

[1] 3GPP TS 23.501 v15.00, System architecture for the 5G system, Stage 2, Dec. 2017.

[2] *View on 5G Architecture*, 5GPPP White Paper v2.0, Dec. 2017.

[3] *Network Functions Virtualization (NFV) Ecosystem Report on SDN Usage in NFV Architectural Framework*, ETSI NFV-EVE White Paper 005, Dec. 2015.

[4] ETSI-NFV, Network Functions Virtualization. [Online]. Available: https://www.etsi.org/technologies/nfv (Accessed: July 14, 2021)

[5] *Network Functions Virtualization* (*NFV*) *Release 2 on Testing;* , *Guidelines on Interoperability Testing for MANO,* ETSI GR NFV-TST 007 v2.6.1, Jan, 2020.

[6] W.P. Lai, H.L. Lai, and M.J. Lai, “A design framework of automatic deployment for 5G network slicing,” *Proc. APSIPA Annual Summit and Conference 2020* (*APSIPA ASC'20*), Auckland, New Zealand, Dec. 2020, pp. 1571–1577.

[7] T. Li, L. Zhao, R. Duan and H. Tian “SBA-based mobile edge computing,” *Proc. IEEE Globecom Workshops 2019 (Globecom'19)*, Waikoloa, Hawaii, USA, Dec. 2019.N. Nikaein et al., “Toward a Cloud-Native Radio Access Network,” in Advances in Mobile Cloud Computing and Big Data in the 5G Era, vol. 22. 3, C. X. Mavromoustakis et al., Ed. Cham: Springer, 2017, pp. 171–202.

[8] M. Patel et al., Mobile-Edge Computing - introductory technical white paper. ETSI White Paper, Sep. 2014.

[9] Heli Zhang, Jun Guo, Lichao Yang, et al., “Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC,” *IEEE Conf. on Computer Communications Workshops*, Atlanta, GA, USA, May 2017, pp. 115–120.

[10] W.P. Lai and K.C. Chiu, “NUMAP: NUMA-aware multi-core pinning and pairing for network slicing at the 5G mobile edge,” *Proc. APSIPA Annual Summit and Conference 2019 (APSIPA ASC'19)*, Lanzhou, China, Nov. 2019, pp. 22–27.

[11] *Description of Network Slicing Concept*, NGMN White Paper, Jan. 2016.

[12] X. Foukas, G. Patounas, A. Elmokashfi and M. K. Marina, “Network slicing in 5G,” *IEEE Comm. Magazine*, vol. 55, no. 5, pp. 94–100, May 2017.

[13] Chia-Yu Chang and Navid Nikaein, “RAN runtime slicing system for flexible and dynamic service execution environment,” *IEEE Access*, vol. 6, pp. 34018–34042, July 2018.

[14] L. E. Li, Z. M. Mao and J. Rexford, "Toward software-defined cellular networks," *Proc. 1st European Workshop on Software Defined Networking* (EWSDN’12), Darmstadt, Germany, 2012, pp. 7–12.

[15] A. Gudipati, D. Perry, L. E. Li and S. Katti, “SoftRAN: Software defined radio access network,” *Proc. ACM SIGCOMM 2nd Workshop on Hot Topics in Software Defined Networking* (HotSDN’13), Hong Kong, China, 2013, pp. 25–30.

[16] A. Gudipati, L. E. Li and S. Katti, “RadioVisor: A slicing plane for radio access networks,” *Proc. ACM SIGCOMM 3rd Workshop on Hot Topics in Software Defined Networking* (HotSDN’14), Chicago, USA, 2014, pp. 237–238.

[17] X. Foukas, N. Nikaein, M. M. Kassem et al., “FlexRAN: A flexible and programmable platform for software-defined radio access networks,” *Proc. 12th International Conference on Emerging Networking Experiments and Technologies* (CoNEXT ’16), Irvine, CA, USA, Dec. 2016, pp. 427–441.

[18] N. Nikaein, C.Y. Chang and K. Alexandris, “Mosaic5G: agile and flexible service platforms for 5G research,” *Comput. Commun. Rev.*, vol. 48, no. 3, pp. 29-34, 2018.

[19] Eurecom, the Mosaic5G project. [Online]. Available: http://mosaic-5g.io/ (Accessed: July 10, 2021)

[20] N. Nikaein, M. K. Marina, S. Manickam et al., "OpenAirInterface: a flexible platform for 5G research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33-38, 2014.

[21] R Schmidt, C.Y. Chang and N. Nikaein, "FlexVRAN: A flexible controller for virtualized RAN Over Heterogeneous Deployments," *Proc. 2019 IEEE International Conference on Communications* (ICC'19), Shanghai, China, 2019.

[22] N. Nikaein et al., "Network store: exploring slicing in future 5G networks," *Proc. 10th Int. Workshop on Mobility in the Evolving Internet Architecture* (*MobiArch '15*), Paris, France, Sep. 2015, pp. 8–13.

[23] 3GPP TS 38.214 v15.3.0 R.15, 5G; NR; Physical layer procedures for data, Oct. 2018.

[24] 3GPP TS 36.213 v13.3.0 R.13, 3GPP; Technical Specification Group Radio Access Network; E-UTRA; Physical layer procedures, Sep. 2016.

[25] Eurecom, the Mosaic5G project's work item, FlexRAN northbound API documentation v2.4.0. [Online]. Available: https://mosaic5g.io/apidocs/flexran (Accessed: July 1, 2021)