5G E2E Network Slicing Predictable Traffic Generator

Brigitte Jaumard
GAIA - Global Artificial Intelligence Accelerator
Ericsson
Montreal (Qc) Canada
brigitte.jaumard@ericsson.com

Junior Momo Ziazet
Computer Science and Software Engineering
Concordia University
Montreal (Qc) Canada
Junior.momoziazet@mail.concordia.ca

***Abstract*—Automated resource management for 5G network slicing implies the need to assign each slice the necessary resources, i.e., the ability to predict their respective requests and resource requirements. Machine learning models and algorithms can meet these needs provided the required data is available. Unfortunately, 5G traffic data remains sparse despite many studies relying on machine learning models and algorithms for traffic forecasting or automated network resource management.**

In this study, we introduce a 5G-type predictable traffic generator that relies on the refactoring of open data of vehicle and pedestrian traffic from the City of Montreal. Indeed, the latter data is refactored in order to generate different classes of network traffic, with different characteristics associated with typical 5G applications, and then with different traffic patterns and peak hours. The result is a valuable traffic generation tool for researchers interested in validating machine learning algorithms aimed at, for example, traffic forecasting, resource elasticity, or automated scaling of slice resources.

***Index Terms*—5G, Network Slicing, Traffic Generator.**

I. Introduction

Network traffic generators are highly valuable tools for evaluating the performance of network management machine learning algorithms, in spite of having access to real data.

In the particular case of 5G network slicing, i.e., a network architecture that allows the multiplexing of virtualized and independent logical networks on the same physical network, it is of the utmost importance to manage the dimensioning of these logical networks efficiently and dynamically, both in terms of network and computer resources to reduce the overall cost of network operation. In order to assign to each slice the necessary resources, it requires the ability to predict their respective demands in order to avoid uniform or reactive resource allocation/dimensioning with respect to peak hours.

Most of the intelligence in 5G networks is done in software, through a set of logical nodes in the 3GPP 5G RAN [1]: radio unit (RU), distributed unit (DU) and centralized unit (CU). These latter logical nodes are connected to the User Plane Function (UPF) of the 5G core network through a series of transport aggregation operations. All together these logical nodes control the traffic flow of the appropriate applications and deliver a set of service functions according to the various service requirements. This led to numerous studies with both traditional optimization/simulation tools and machine learning models and algorithms, for the placement of these logical nodes, their auto-scaling with elastic orchestration [2], while there is a lack of appropriate data, of significant sizes, to validate these algorithms and compare their performance.

While some 5G RAN traffic simulators have been recently developed, see, e.g., Corcoran *et al.* [3], Nardini *et al.* [4], they lack generalities and do not allow the comparison of 5G E2E provisioning algorithms. Due to the lack of available 5G traffic data sets, current studies on traffic prediction or on resource orchestration use either synthetic data, Guo *et al.* [5], or very limited real data, Cappanera *et al.* [6], see also, e.g., [7], [8], or data that is not 5G, Selvi and Thamilselvan [9], which limits the validation and performance study of machine learning algorithms.

Our contribution lies in designing and developing a dynamic set of 5G demands, with changing traffic patterns and intensities over time. For this, we use the open data of the City of Montreal [10], which is real traffic data.

The paper is organized as follows. In Section II, we provide an overview of the various studies aimed at overcoming the lack of real data to test machine learning algorithms for the automated management of 5G networks. In Section III, we briefly recall the background of a 5G traffic generator and its networking environment. In Section IV, we describe the source of real data that we use for our 5G traffic generator. In Section V, we describe the traffic generation scheme. In Section VI, we provide several results of the 5G Traffic Generator, to illustrate its functionality and versatility, and its suitability for testing 5G provisioning and elastic E2E user plane flow in different dynamic traffic scenarios.

II. Literature Review

Even though there are few 5G traffic simulators today, real 5G traffic data is still difficult to access, especially in terms of traffic forecasting. Currently commonly used 5G simulators are Simu5G [4], [11], However, none of these simulators offer the option of 5G predictable traffic, i.e., generating 5G traffic on which we can build traffic forecasts, since there are no traffic patterns in the traffic generator.

Although there are several studies on 5G traffic forecasting, few of them use actual 5G traffic (see for example) and when this is the case, the data is very limited. Often traffic generators are tested on other datasets, see for example Yan *et al.* [12]

978-3-903176-59-1

who tested their 5G traffic generator on, for example, weather data.

Since 5G data traffic is very rare, limited to very specific traffic classes and over very limited periods of time, some studies focus on 5G traffic generation. For example, Kim *et al.* [13] propose a 5G neural traffic generation model and a methodology to calculate the spectrum requirements of private 5G networks to provide various industrial communication services.

In this study, we extend the work of Ziazet *et al.* [14] in order to generate E2E traffic with the modelling of the sequence of logical nodes (RU, DU, CU, UPF), so as to provide meaningful data sets for testing, e.g., automated scaling of compute resources.

III. 5G Networking and Orchestration Environment

We now describe the key elements of the 5G network slicing environment for the development of a traffic generator.

A. 5G E2E Network Slicing Environment

We consider a reference 5G network environment characterized by three segments: Radio Access Network, Transport Network and Core Network, see Figure 1. We add to this the network slicing component, i.e. the ability to simultaneously deploy and use different dedicated virtual networks, each specialized in the provision of a given set of services and/or a set with given subscribers. In this context, software-defined networking (SDN) and network function virtualization (NFV) technologies play a critical role in the design of 5G network slices.

Slices are virtual networks composed first of a 5G E2E user plane flow and then, in the data network (DN) of one or more ordered virtual network functions (VNFs), defining a so-called service function chain (SFC). Example of a SFC, e.g., for VoIP, is NAT $\rightarrow$ FW $\rightarrow$ TM $\rightarrow$ FW $\rightarrow$ NAT, with the following VNFs: Network Address Translation (NAT), Firewall (FW), and Traffic Monitor (TM), see [15].

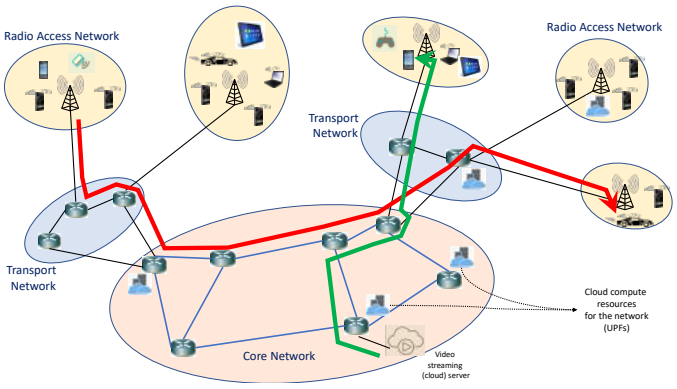


Fig. 1. Reference 5G E2E Network Slicing Environment

The 3GPP 5G RAN architecture [16] consists of a set of radio base stations (known as gNBs) connected to the 5G core network and to each other. The gNB includes three main functional modules: a Radio Unit (RU), a distributed unit (DU), responsible for real time scheduling functions, and a centralized unit (CU) responsible for non-real time ones. In a 5G cloud RAN, the DU's server and relevant software could be hosted on a site itself or can be hosted in an edge cloud (datacenter or central office). The CU's server and relevant software can be co-located with the DU or hosted in a regional cloud data center. In 5G RAN, at the network level, we distinguish three parts: Fronthaul, the link connectivity between the RU and DU ; Midhaul, the link connectivity between the DU and CU ; and lastly the Backhaul, the link connectivity between the CU and the core network.

In the core network, the User Plane Function (UPF) performs user processing and transfers data. While control plane functions can be shared between network slices, user plane functions (UPFs) are slice specific in terms of QoS requirements.

Although a 5G network is often built as an assembly of different components and specialized networks, an E2E vision makes it possible to better understand the requirements of the 5G network and to provide more efficient 5G solutions to satisfy users' quality of service requirements and to meet operators' business needs.

In the sequel, we model the logical 5G network as a directed graph $G = (V, L)$, where V is the set of nodes and L is the set of logical links. A subset of the nodes is equipped with computer resources (e.g., servers for applications such as gaming/video streaming or datacenters for hosting UPFs), and are commonly called compute nodes. These nodes have processing capabilities in terms of VMs or containers where each VM/container is characterized by its number of CPUs (and their vCPU counts), RAM and storage.

B. 5G Provisioning

A 5G request is characterized by a source and a destination in the logical network, start-up and hold time, bandwidth requirement, end-to-end delay, and the required service function chain (SFC). The source or destination can be the location of a UE or the source (server location) of, e.g., a downstream video stream. Latency of a request has two components: *(i)* the network component with its four parts, i.e., propagation, transmission, queuing delay, and processing delays, both in the core network (CN) and in the Radio Access Network (RAN). *(ii)* the software component with the processing times associated with the various 5G logical functionalities. We will

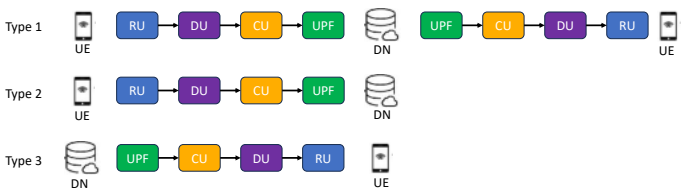


Fig. 2. Three types of 5G user plane flows

consider three types of 5G user plane flows (downlink, uplink,

bidirectional) as described in Figure 2. We will not go through the details of the functions embedded in each logical functionality, with the understanding that these functions perform according to the QoS requirements. Each entity, RU, CU, DU and UPF has its own hardware (RU)/software (CU, DU, UPF) requirements. Software requirements are translated into compute requirements, which vary with the class of traffic, in terms of CPU, RAM and storage resources. The latter ones are provided by the virtual machines (VMs) or the virtualized containers hosting the 5G logical functionalities.

IV. A Traffic Generator with a Live Traffic Data Source

Over time, for example over a day, 5G traffic patterns change, both in terms of overall distribution and intensity, but also in terms of the nature of applications. While we see many work today with traffic prediction and elastic resource provisioning, there is a lack of data sets to test and validate the proposed algorithms. In this work, we attempt to provide a versatile traffic generator that can fill that gap.

A. Generator Need and Motivation

NFV and SDN technologies provide efficiency in controlling the traffic and elasticity in deploying and scaling up/down in/out network and compute resources. Indeed, provisioning new resources is made easier as it is a matter of software deployment. However, network operators are in need of reducing energy consumption in the future B5G networks, in addition to the traditional OPEX/CAPEX cost minimization. For this reason, elastic orchestration is a critical issue, with the definition of an optimized strategy for the resources to be made available when needed (when to scale and how much to scale) remaining complex. Note that resource elasticity is indeed an ITU requirement [17].

B. Open Data Sets of the City of Montreal

In the sequel, we will explain how we re-use some of the open data sets of the City of Montreal for our 5G-type traffic data. We next describe briefly these data.

Among the open data of the City of Montreal [10], we used the traffic data with the counts of different types of vehicles, bicycles and pedestrians at a given number of street intersections. Measurements are taken at 15 minute intervals during certain times of the day and data is available from 2008 to present.. The observations detail the start time of each count period, the number, origin and direction of vehicles, pedestrians and cyclists for each possible movement at an intersection and the geographical coordinates of the intersection. The counted entities are classified into sixteen categories: Cars, Light Trucks, Heavy Trucks, Pedestrians, Bicycles, Buses, Schoolchildren, Trucks, Straight trucks, Articulated trucks, Motorcycles, Unused, Uturn, Illegal, Other and All.

C. Different Traffic Patterns

To generate non-uniform and meaningful traffic distribution, we adapted the population gravity model used in [18] for various transport network traffic scenarios, with a refactoring (see Section V-D for the details) of the open vehicular and pedestrian data of the city of Montreal [10].

The number of service requests per node pair is proportional to the product of the respective populations divided by the distance between them. Indeed, at time t, each network node $v \in V$ has a population (aka users) denoted by $N_v(t)$ and the traffic of node pairs $(v, v') \in V \times V$ is computed as follows:

$$P_{vv'}(t) = \frac{\log(100 + \frac{N_v(t)N_{v'}(t)}{D_{vv'}})}{\sum_{w \in V} \sum_{w' \in V} \log(100 + \frac{N_w(t)N_{w'}(t)}{D_{ww'}})}, \qquad (1)$$

where $D_{vv'}$ is the geographical distance between nodes v and v', $v \neq v'$.

In order to allow requests within the same RAN, i.e., requests with the same source/destination node in the core network, we define D_{vv} as a scaling factor rather than a distance, and decompose $N_v(t) = N_v^{\text{IN}}(t) + N_v^{\text{OUT}}(t)$ for the number of users with service requests within the same RAN, and between two different RANs, respectively. As a consequence, $P_{vv'}(t)$ is computed with $N_v^{\text{OUT}}(t)$ for $v \neq v'$, and with $N_v^{\text{IN}}(t)$ when $v = v'$. Note that the addition of the $\log$ factor is motivated by the need to smooth the significant unbalance among node data.

V. Proposed 5G Traffic Generator

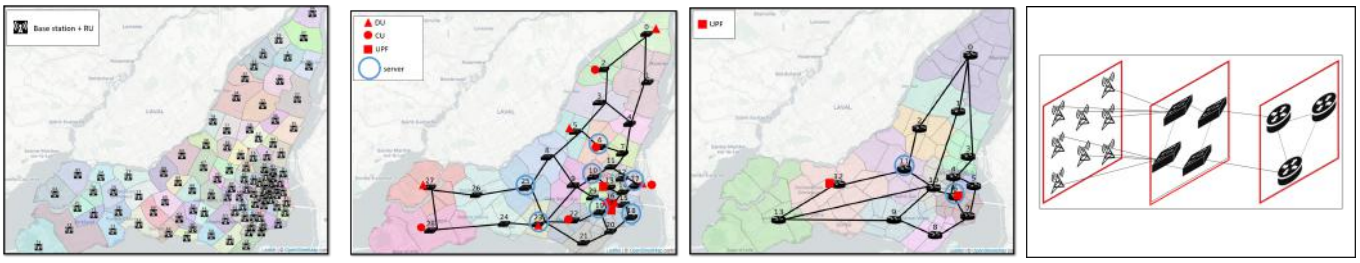
A. Network Environment

We design the logical layer of a E2E network relying on open data from the urban traffic data of the City of Montreal [10]. It contains the traffic counting information of different categories of vehicles every 15 minutes at different traffic light intersections of the city of Montreal. Based on the geographical locations of the intersections, we clustered them into 100 cells, representing the radio base stations, see Figure 4(a). The base stations are connected to the transport network shown in Figure 4(b), which in turn is connected to the core network represented in Figure 4(c). Each base station node network is abstractly associated with a set of UEs, each of which is associated with one or several types of applications and its typical access delay. RUs are hosted at the base station, while DUs and CUs reside in the edge cloud of the transport network.. To accommodate latency-sensitive applications (e.g., URLLC slices), some UPFs are hosted in the edge cloud while others are located in the central cloud. The resulting network is shown in Figure 4, where some servers running applications such as gaming/video streaming are depicted. Figure 4(d) illustrates how the different network components are connected.

B. Sequence of Logical Nodes and VNFs

We considered 6 types of services and their corresponding sequence of logical nodes. Behind every logical nodes, there is a sequence of one or several VNFs that we do not detail. We reuse some of the latency and bandwidth requirements reported in [15]. We adapted some values, in particular those of Massive IoT (MIoT), using, e.g., [19]. We provide the typical

Fig. 3. 5G network: network and service providers infrastructures



(a) Radio network (b) Transport network (c) Core network (d) Network hierarchy

Fig. 4. 5G network: radio, transport and core infrastructures and their hierarchy

E2E requirements, with augmented reality and Industry 4.0 (smart factory) sharing the most stringent ones, see, e.g., [20]. Table II provides the compute resource modeling that we use with a required amount of CPU (in terms of percentage of CPU per user), RAM and Storage for each logical node. These values do not come from real use cases (due to lack of access to real data) and only provide a certain order of magnitude.

TABLE I
5G USER PLANE FLOWS (BANDWIDTH AND LATENCY VALUES ADAPTED FROM [15])

Services	E2E user plane flows	Bandwidth per user or IoT system	E2E latency	Request bundles
Cloud gaming	UPF_{CG} - CU - DU - RU	4 Mbps	80 ms	[40-55]
Augmented reality	UPF_{AR} - CU - DU - RU	100 Mbps	10 ms	[1-4]
VoIP	RU - DU - CU - UPF_{VOIP} - UPF_{VOIP}- CU - DU - RU	64 Kbps	100 ms	[100-200]
Video streaming	UPF_{VS} - CU - DU - RU	4 Mbps	100 ms	[50-100]
Massive IoT	RU - DU - CU - UPF_{MIOT}	[1-50] Mbps	5 ms	[10-15]
Industry 4.0	RU - DU - CU - $UPF_{I4.0}$ - $UPF_{I4.0}$ - CU - DU - RU	70 Mbps	8 ms	[1-4]

TABLE II
CPU/RAM/STORAGE CORE USAGE FOR 5G LOGICAL FUNCTIONALITIES

5G logical functionalities	vCPU	RAM (Gb) per 100 Mbps	Storage (Gb) per 100 Mbps	5G Logical functionalities processing time per 0.01 msec unit
RU	1	4	7	12
DU	9	5	1	6
CU	11	15	2	22
UPF_{CG}	13	15	7	14
UPF_{AR}	5	2	5	16
UPF_{VOIP}	5	2	5	2
UPF_{CG}	5	11	10	4
UPF_{MIOT}	5	3	20	4
$UPF_{I4.0}$	5	4	11	4

We considered 6 different slices as illustrated in Table I with different bandwidth and latency requirements.

C. Network Dimensioning

Another important aspect is the dimensioning of the network links, as it impacts the provisioning of the service requests, their access to compute resources and then the end-to-end delay. Having in mind that most of the network operators have enough capacity to grant most of the requests if not all on heavy traffic periods (peak times) and that the transport capacities are set to last for a particular duration (e.g., a few weeks to a few months), we dimensioned the links and compute nodes capacities in such a way that GoS is acceptable even in heavy traffic periods. Indeed, we use data from the busiest traffic periods of the city of Montreal to dimension the network. Starting with a network with unbounded link capacity and compute node resources, we routed all the busiest period generated data on the shortest path and recorded for each link and compute node their maximum resource usage. The shortest path was run on the multi-layer graph [21] that is commonly used for the provisioning of 5G service requests with SFCs, with links weighted by network delays and cross-layer links weighted by the 5G logical functionalities processing times. We then set each transport capacity value to a number uniformly selected from the range [90, 110]% of its corresponding maximum usage within the shortest path usage. This is to enforce that the shortest path will not always be used.

Regarding the E2E delay, we defined it carefully so that we get a reasonable delay for each application and request. Tables I and II provide the resulting values.

D. Request Generation for a given distribution

As explained in Section IV-C, we refactor the data of the City of Montreal in order to use them to simulate different 5G slices, each with a different traffic pattern and intensity over the days. The resulting slice traffic, although not necessarily representative of the associated application, corresponds to real data, i.e., a good fit for validating and testing machine learning algorithms for, e.g., traffic prediction or elastic resource management.

Considering the urban traffic of the city of Montreal, we constructed 6 artificial slices (sequences of logical nodes in the context of a 5G network) by combining some urban traffics as presented in the table III. For each slice, we used the overall count of the associated vehicles/pedestrian at the street intersections at specific time stamps. Since the statistics are collected every 15 minutes in [10], we end up with a dynamic vehicle/pedestrian model which, in turn, gives a dynamic and non-uniform traffic distribution, which

is artificial as far as 5G applications are concerned, but still with real data behavior. The pattern of the original urban traffic data being different from the one of network traffic data like represented in, e.g., [22] (gaming), [23] (video streaming) for different applications, we used some scaling, shifting and transformation functions to change the urban traffic data and obtain a similar pattern with the network data. This way, we got organized so that video streaming (Slice 0), which is today the dominant traffic, has a large share of the traffic with peak hours late in the evening. Indeed, studies show that video traffic account for about 70 % in 2022, a share that is forecast to increase to about 80 % in 2027.

TABLE III
MAPPING OF URBAN TRAFFIC TO 5G SLICES

Service chains	Slices	Vehicle/Pedestrian types
Video streaming	Slice 0	Cars
Cloud Gaming	Slice 1	Pedestrians + Schoolchildren
VoIP	Slice 2	all Trucks categories
MIoT	Slice 3	Bikes + Motorcycles
Industry 4.0	Slice 4	Buses
Augmented reality	Slice 5	all other categories

Using the data mapping described in Table III, we first get a number of requests per slice using the number of items, i.e., vehicles or pedestrians. We then derive an estimate of the required bandwidth for each slice at each time period (every 15 minute) with the product of the number of user requests and the corresponding bandwidth BW^{SLICE} requirement associated with each slice. Using the gravity model that was described in Section IV-C, we next non uniformly distribute the traffic requests among the different node pairs, using the user scaling given in the last column of Table I, i.e., each request is now a bundle of user requests.

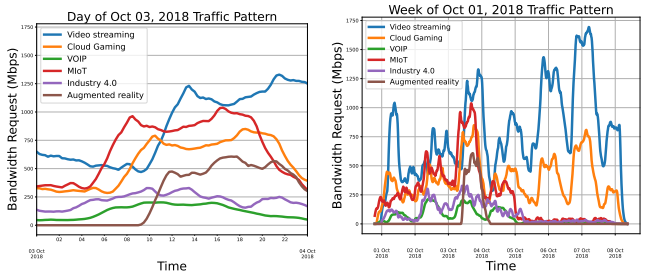
Another vital aspect of our dynamic traffic generator is to set the start t_i and end t'_i times of a request i (holding time $h_i = t'_i - t_i$). We generate or terminate a request based on the gravity model. Using the transformed data from the city of Montreal, we know for each time period the number of requests to generate per slice. The start time t_i of request i is then given by the transformed data of the city of Montreal. The holding time of a request h_i was selected using a geometric distribution with a mean of 5, from which the end time was then inferred. Each generated request is an aggregate of user-requests with the same application, the corresponding number of users per request was randomly selected in the range defined in the last column of Table I.

VI. CHARACTERISTICS AND ANALYSIS OF THE GENERATED TRAFFIC DATA

We present here some characteristics of the generated traffic data.

A. Different weekly patterns depending on the applications

An advantage of using our generator is that, as it is based on open data from the city of Montreal which is updated every 15 minutes, we can generate meaningful data from any selected time period with a very diverse load. Figure 5 presents the generated traffic pattern of each application of the week going from October 1st to October 7th, 2018. We observe that the traffic varies globally according to the days and the applications, video streaming and cloud gaming being the dominant applications with more than 50% of the traffic on the one hand and the least bandwidth-intensive VoIP on the other hand. Video streaming, cloud gaming and augmented reality have their peak at similar times, while different behaviours are observed for VoIP, Industry 4.0 and MIoT. Observed fluctuations are related to a real event (urban mobility), and knowing that in 5G, user mobility is an important aspect, the generated data can therefore help to train a machine learning model, when we expect a change in pattern and distribution.



(a) One daily traffic pattern (b) One weekly traffic pattern

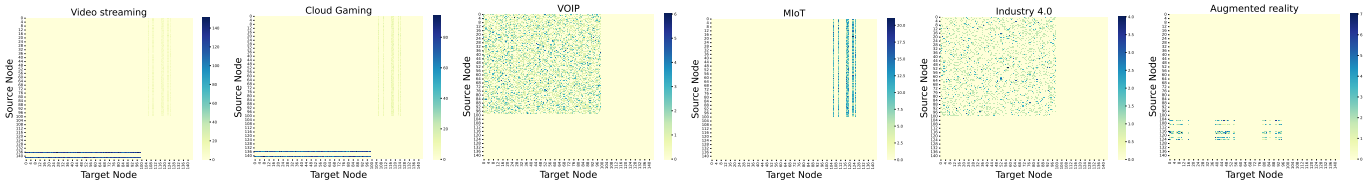
Fig. 5. Traffic patterns

B. Heat maps

To visualize the distribution of the requests, Figure 6 illustrates the number of requests per node pair within the network. The illustration has been done per application, as they do not necessarily share the same source and target set. We can easily identify the three types of applications, server-to-base station downlink based applications where the source nodes are servers and the target ones are any base station node; base station-to-server uplink based applications where the source nodes are base stations and the target ones are servers; and finally anyone-to-anyone uplink and downlink based applications where the source/destination nodes can be any base station. Overall, we can see that the distribution of traffic is not uniform and is spread over the network nodes. All applications have a non-uniform distribution of demand based on their respective sets of sources and targets. Augmented reality applications have been located in the heart of the city of Montreal in our settings. Therefore, their corresponding distributions are only distributed on nodes located in the downtown area.

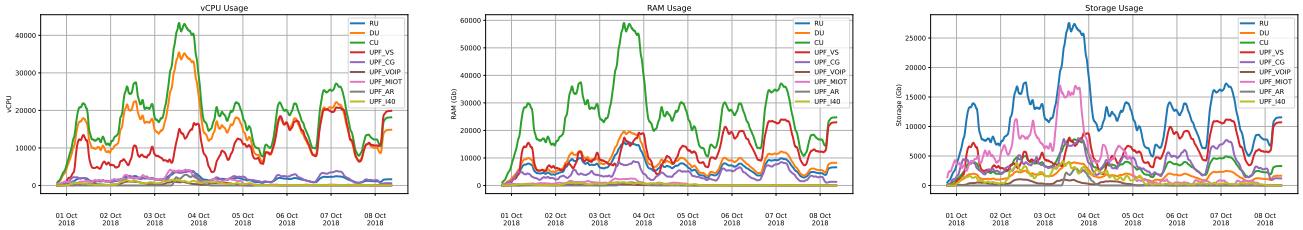
C. Compute resources

To set a proper initial dimensioning of the compute resources based on the generated data, we used the shortest available weighted path and measured the resource utilisation for each service request. Figure 7 presents the amount of compute resources per logical node over the selected period. As expected, we can see a correlation between resource



(a) Video Streaming (b) Gaming (c) VoIP (d) MIoT (e) Industry 4.0 (f) Augmented Reality

Fig. 6. Heat maps of the bandwidth requirements of the requests



(a) CPU (b) RAM (c) Storage

Fig. 7. Compute resources over one week per logical node

consumption and traffic load. Indeed, more resources were used on Oct 3rd as it was the day with the most traffic in the data of the city of Montreal.

The observed utilization patterns are highly dependent on the defined parameters. In spite of being able to find practical values for the various resource consumption, we define them arbitrarily, with linear consumption with respect to the bandwidth, while some of them maybe nonlinear [24]. However, we set the values differently for the different UPFs to reflect the diversity of these values depending on the type of services. This usage is presented here to show that we can train models with our datasets to get some proof of concept.

D. Path delay vs. delay Requirement

Delay requirements by application are illustrated in Figure 8 where we plot the cumulative distribution function (CDF) of path delay versus application delay requirement. The processing delay of the various logical nodes is made proportional to the bandwidth requirements of the service requests. We observe that for applications with longer E2E, the overall delay is much smaller in % than for the applications with more stringent delays such as MIoT, Industry 4.0 or Augmented Reality. Applications with stringent delays are the ones most likely to encounter delay issues, with, e.g., MIoT suffering from rejection of requests due to unmet delay requirements.

E. Grade of Service (GoS)

In Figure 10 we present the grade of service computed as the ratio of the throughput over the offered load. We observe that MIoT starts to decrease on Oct. 1st and experiences the highest denial rate with a GoS around 84%. Augmented reality starts decreasing towards the middle of Oct. 3^{rd}, which represents the peak traffic time that was considered to dimension the network. The overall aggregated GoS, see the dashed line, is above 90%, indicating the quality of the proposal.

VII. Conclusion

We designed an enhanced 5G E2E traffic generator relying on live data, with meaningful data for testing and evaluating machine learning algorithms in relation with traffic prediction or management of elastic orchestration.

Acknowledgment

Work of the second author was supported by a NSERC/INNOVEE internship in collaboration with Ericsson. We would also like to thank Adel Larabi for sharing his 5G expertise and answering all our technology questions.

References

[1] 3GPP, "System architecture for the 5G System (5GS)," ThreeGPP, Technical Specification (TS) 23.501, 12 2021, TS 23.501, Version 17.3.0.

[2] L. Khan, I. Yaqoob, N. Tran, Z. Han, and C. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36 009–36 028, 2020.

[3] D. Corcoran, P. Kreuger, and C. Schulte, "Efficient real-time traffic generation for 5G RAN," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Budapest, Hungary, 2020, pp. 1 – 9.

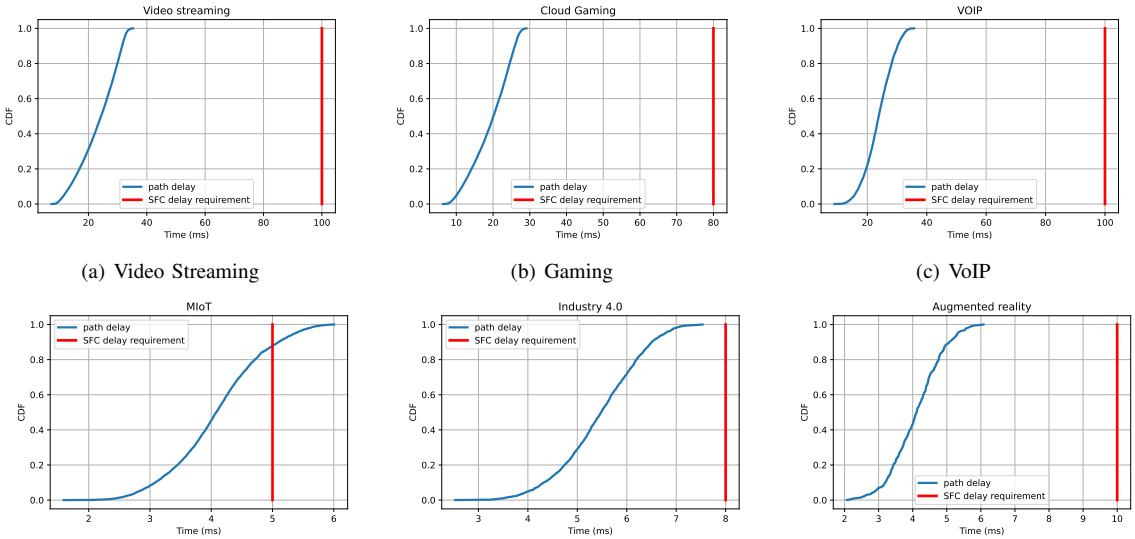
[4] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G – an OMNeT++ library for end-to-end performance evaluation of 5G networks," *IEEE Access*, vol. 8, pp. 181 176 – 181 191, 2020.

[5] Q. Guo, R. Gu, Z. Wang, T. Zhao, Y. Ji, J. Kong, R. Gour, and J. P. Jue, "Proactive dynamic network slicing with deep learning based short-term traffic prediction for 5g transport network," in *Optical Fiber Communication Conference - OFC*, 2019, pp. 1–3.

[6] P. Cappaneraa, F. Paganelli, and F. Paradisoa, "VNF placement for service chaining in a distributed cloud environment with multiple stakeholders," *Computer Communications*, vol. 133, pp. 24 – 40, 2019.

[7] M. Chen, Y. Miao, H. Gharavi, L. Hu, and I. Humar, "Intelligent traffic adaptive resource allocation for edge computing-based 5G networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 499–508, 2020.

[8] T. D. Tran, B. Jaumard, H. Duong, and K.-K. Nguyen, "Joint service function chain embedding and routing in cloud-based nfv: A deep q-learning based approach," in *2021 IEEE 4th 5G World Forum (5GWF)*, 2021, pp. 171–175.



(a) Video Streaming (b) Gaming (c) VoIP

(d) MIoT (e) Industry 4.0 (f) Augmented Reality

Fig. 8. CDF - Path delay vs delay requirements

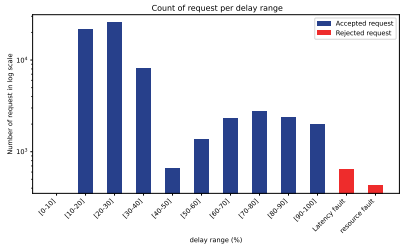


Fig. 9. Count of requests per delay range, accepted vs. rejected.

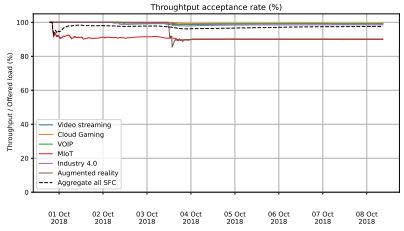


Fig. 10. Grade of Service (GoS)

[9] K. T. Selvi and R. Thamilselvan, "An intelligent traffic prediction framework for 5G network using SDN and fusion learning," *Peer-to-Peer Networking and Applications*, vol. 15, pp. 751 – 767, 2022.

[10] "Comptages des véhicules, cyclistes et piétons aux intersections munies de feux de circulation," https://donnees.montreal.ca/ville-de-montreal/comptage-vehicules-pietons, accessed: 2022-04-06.

[11] G. Nardini, G. Stea, A. Virdis, and D. Sabella, "Simu5G: a system-level simulator for 5G networks," in *SIMULTECH*, July 2020, pp. 10 903–10 924.

[12] Y. Yang, S. Geng, B. Zhang, J. Zhang, Z. Wang, Y. Zhang, and D. Doermann, "Long term 5G network traffic forecasting via modeling non-stationarity with deep learning," *Communications Enginering*, vol. 2, no. 33, pp. 1 – 12, 2023.

[13] D. Kim, M. Ko, S. Kim, S. Moon, K.-Y. Cheon, S. Park, Y. Kim, H. Yoon, and Y.-H. Choi, "Design and implementation of traffic generation model and spectrum requirement calculator for private 5g network," *IEEE Access*, vol. 10, pp. 15 978 – 15 993, 2022.

[14] J. Ziazet, B. Jaumard, H. Duong, P. Khoshabi, and E. Janulewicz, "A dynamic traffic generator for elastic 5G network slicing," in *IEEE International Symposium on Measurements & Networking (M&N)*, 2022, pp. 1–6.

[15] L. Askari, A. Hmaity, F. Musumeci, and M. Tornatore, "Virtual-network-function placement for dynamic service chaining in metro-area networks," in *International Conference on Optical Network Design and Modeling (ONDM)*, Dublin, Ireland, 2018, pp. 136 – 141.

[16] 3GPP, "5G - Procedures for the 5G System," European Telecommunications Standards Institute (ETSI), Technical Specification (TS) 23.502, 2018, 3GPP TS 23.502 version 15.2.0 Release 15.

[17] ITU-T, "M.3400. TMN management functions," https://datatracker.ietf.org/doc/draft-eastlake-sfc-parallel/, 2000.

[18] A. Betker, C. Gerlach, R. Hülsermann, M. Jäger, M. Barry, S. Bodamer, J. Späth, C. Gauger, and M. Köhn, "Reference transport network scenarios," BMBF Multi Tera Net, Tech. Rep., July 2003.

[19] Y. A. Mtawa, A. Haque, and B. Bitar, "The mammoth internet: Are we ready?" *IEEE Access*, vol. 7, pp. 132 894–132 908, 2019.

[20] J. Walia, H. Hämmäinena, K. Kilkki, and S. Yrjölä, "5G network slicing strategies for a smart factory," *Computers in Industry*, vol. 111, pp. 108 – 120, October 2019.

[21] N. Huin, B. Jaumard, and F. Giroire, "Optimal network service chain provisioning," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1320–1333, June 2018.

[22] P.-Y. Tarng, K.-T. Chen, and P. Huang, "An analysis of WoW players' game hours," in *7th ACM SIGCOMM Workshop on Network and System Support for Games, NETGAMES*, Worcester, Massachusetts, USA, October 2008, pp. 1 – 7.

[23] S. Rahman, H. Mun, H. Lee, Y. Lee, M. Tornatore, and B. Mukherjee, "Insights from analysis of video streaming data to improve resource management," in *IEEE 7th International Conference on Cloud Networking (CloudNet)*, 2018, pp. 1–3.

[24] S. V. Rossem, W. Tavernier, D. Colle, M. Pickavet, and P. Demeester, "Profile-based resource allocation for virtualized network functions," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1374–1388, 2019.