Big Data for 5G Intelligent Network Slicing Management

Hatim Chergui and Christos Verikoukis

ABSTRACT

Network slicing is a powerful tool to harness the full potential of 5G systems. It allows verticals to own and exploit independent logical networks on top of the same physical infrastructure. Motivated by the emergence of the big data paradigm, this article focuses on the enablers of big-data-based intelligent network slicing. The article starts by revisiting the architecture of this technology that consists of data collection, storage, processing, and analytics before it highlights their relationship with network slicing concepts and the underlying trade-offs. It then proposes a complete framework for implementing big-data-driven dynamic slicing resource provisioning while respecting SLAs. This includes the development of low-complexity slices’ traffic predictors, resource allocation models, and SLA enforcement via constrained deep learning. The article finally identifies the key challenges and open research directions in this emerging area.

INTRODUCTION

With the emergence of 5G ultra-dense networks, the collected amount of performance data has exploded, and the integration of new technologies to handle and analyze it has become necessary. This task is relevant to the realm of big data, where usually the datasets are beyond the ability of typical database software tools to capture, store, and analyze [1]. Hence, a great challenge for 5G systems management lies in the integration of big data analysis tools into the key performance indicator (KPI) platforms to efficiently manage the available network resources.

Network slicing is also a key feature in 5G cellular systems. It enables running fully or partly isolated logical networks on the same physical network, thereby offering a substantial resource multiplexing gain [2]. Each end-to-end logical network — or slice — is owned by a logical operator called a tenant and managed by the physical operator according to an agreed service level agreement (SLA). By leveraging technologies like software defined networking (SDN)/network function virtualization (NFV), network softwarization yields the programmability and flexibility to create tailored and self-contained slices on top of the same physical infrastructure. This allows the coexistence of several verticals over the 5G network.

The full isolation of slices, however, may present a high cost in terms of efficiency. Therefore, network slicing should be combined with solutions for dynamic orchestration of resources [3]. In this context, the advent of machine learning (ML) techniques and in particular deep neural networks (DNNs) is expected to be the cornerstone in the automation of end-to-end resource provisioning for intelligent network slicing. This includes schemes for slices’ traffic prediction such as long short-term memories (LSTMs) and gated recurrent units (GRUs)[4], to name a couple, as well as standard DNNs to model and estimate the required resources at each network function.

To fully exploit the potential of deep learning algorithms, large datasets are required for training the underlying models. Such big datasets are usually created and properly formatted by the network operation subsystem (OSS). This makes their storage, processing, and analysis a great challenge, especially when dealing with 5G ultra-dense networks where the collected data is voluminous and even unstructured. Due to the limitations of the traditional data handling approaches, the aforementioned deep learning task should be viewed as a big data exercise that can be tackled through the technologies and architectures developed in the field. In particular, advanced file systems, data storage and retrieval, and distributed and parallel processing techniques might be invoked.

On the other hand, intelligent slicing management involves the control of the SLA for each slice. This consists of imposing thresholds on some KPIs — agreed between the tenant and the physical operator — and making sure that their violations do not exceed some preset upper bounds. With this intent, the optimization of the big-data-driven machine learning models has to take into account these constraints, which is a challenging task given that the SLA violation rate, for instance, is usually a non-convex non-differentiable function.

To explore the aforementioned aspects, the article starts by highlighting the big data collection, storage, and processing technologies that might be used for intelligent network slicing. The article then revisits network slicing concepts and their link with the big data paradigm while showcasing the underlying trade-offs. Finally, it proposes a complete framework for implementing big-data-based dynamic network slicing, including low-complexity traffic prediction, resource provisioning, and SLA enforcement using constrained deep learning.

BIG-DATA-DRIVEN INTELLIGENT SLICING ARCHITECTURE

An emerging solution for service-oriented networking is network slicing, which creates different and independent slices over the same...
physical infrastructure, spanning from the access domain to the core network domain. A network slice is a set of network resources for a given application or use case. It can be customized to meet the corresponding end-to-end service requirement such as latency and reliability. The key for network slicing is to partition network-wide heterogeneous resources for different slices to support various use cases efficiently. Therefore, it is important but challenging to efficiently utilize network resources while satisfying the service requirements of various applications or use cases.

Figure 1 depicts a typical architecture of big-data-enabled network slicing. It consists of a closed loop that starts by collecting KPI data that are then stored and preprocessed using specific big data technologies to yield structured log files. Finally, the data is analyzed using analytics tools, such as machine learning algorithms, to control the amount of resources allocated to each slice. These building blocks are addressed later.

**Big Data Collection for 5G Network Slicing**

In 5G cellular networks, the OSS is responsible for end-to-end performance, traces, alarms and inventory data collection, and monitoring. It also performs the formatting of the collected big data into structured log files that can be exploited by third party applications. By endowing the OSS with performance probes, it becomes possible to distinguish the traffic of each over-the-top (OTT) application down the network slices accordingly. Through performance counters aggregation, one may get the KPIs of each slice separately. On the other hand, the granularity of the collected performance data usually ranges from 15 minutes for a period of 24 hours to one month for a period of one year. These limitations stem from the storage and processing constraints of the performance data servers. For instance, the hourly granularity can collect the data for 30 days only. Otherwise, the obtained log files become voluminous and difficult to process and exploit by the other network functions. The emergence of intelligent massive network slicing would generate further big OSS data as well as the need to access large datasets to, for example, train slices’ orchestration machine learning algorithms. Hence, to implement dynamic network slicing in practice, it is necessary to seek more efficient ways to store and process the OSS data without imposing strict constraints on its volume, given that long-term performance data is required to understand the underlying patterns.

**Big Data Storage and Processing**

To handle network-wide performance big data resulting from the creation of a large number of slices, several storage and processing technologies exist. The Apache Hadoop software library, for instance, is a framework that allows the distributed processing of large datasets across clusters of servers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. On the other hand, Apache Spark [5] is a unified computing engine and a set of libraries for parallel data processing on computer clusters. At the time of this writing, Spark is the most actively developed open source engine for this task, making it the de facto tool for big data parallel processing. Spark includes libraries for diverse tasks ranging from SQL to streaming and machine learning, and runs anywhere from a laptop to a cluster of thousands of servers. This makes it an easy system to start with and scale up to big data processing on an incredibly large scale. Hadoop reads and writes files to the so-called Hadoop data file system (HDFS). Spark processes data in the random access memory (RAM) using a concept known as resilient distributed dataset (RDD). Spark can run in standalone mode, with a Hadoop cluster serving as the data source. In this case, Spark can be deployed directly on HDFS, and resources can be statically allocated on all or some machines in a Hadoop cluster.

**Big Data Analytics for Network Slicing**

On top of the big data storage and distributed processing frameworks, several data analytics might be used to control resource allocation for network slicing. Hence, recurrent neural network (RNNs) variants such as LSTM and gated recurrent unit (GRU) can be adopted to predict traffic, resource usage, and even future network events. In this respect, many ML architectures have been proposed in the literature. In [6], for instance, convolutional neural networks (CNNs) have been used as slices’ traffic predictors, but at the expense of high complexity [7]. In [8], the authors used the Holt-Winters forecasting procedure to analyze and predict future traffic requests associated with a particular network slice. Such a system, however, is hard to tune and scale, and it is difficult to add exogenous variables [9]. On the other hand, [10] proposed a spatio-temporal neural network (STN) architecture purposely designed for concise network-wide mobile traffic forecasting. It also presented a mechanism that fine-tunes the STN and enables its operation with only limited ground truth observations. The obtained traffic predictions, however, do not exactly match the measured data. By leveraging predictions, proactive resource allocation and corrective actions can be performed in advance to avoid slice faults or service failures. By doing so, network reliability can be significantly improved without much manual effort for maintenance.
Isol At Ion vs. bIg-dAtA-drIven  resource  Alloc At Ion

The development of NFV [11] greatly facilitates the implementation of network slicing. Indeed, NFV enables network functions to be created in virtualized environments rather than on dedicated hardware platforms, thereby significantly improving network scalability and flexibility. With NFV, a number of virtualized network functions (VNFs) (e.g., baseband processing, traffic splitting, data aggregation, deep packet inspection) that the data stream needs to go through can be composed. Also, NFV introduces an additional degree of freedom regarding the placement of these functions in the network. Intelligent placement may improve network performance and reduce operating costs. From a technical standpoint, solutions that implement NFV at different network levels are well established, and have started to be tested and deployed. Examples include current management and orchestration (MANO) platforms architectures like European Telecommunications Standards Institute (ETSI) NFV, and implementations such as OSM, which allow reconfiguring and reassigning resources to VNFs on the fly. In contrast, the flexibility of VNF locations presents a great challenge to the management of network slicing. Indeed, centralized machine learning algorithms that automate, for example, end-to-end resource allocation, require big datasets to be collected and forwarded to a central function, which results in the exchange of large data volumes with all the delays that this incurs. By adopting distributed ML algorithms, the big data transfer burden can be alleviated but at the expense of a centralized optimal decision.

Isol At Ion vs. bIg-dAtA-drIven  resource  Alloc At Ion

Isolation is a major prerequisite that must be satisfied to operate parallel slices on a common physical network. This includes end-to-end performance isolation, wherein the target KPIs of a given slice should always be met regardless of the performance status of other slices. In particular, the assigned resources such as physical resource blocks (PRBs) must be subject only to the dynam-

TRADE-OFFS OF
BIG-DATA-ENABLED NETWORK SLICING

Virtualization vs. Big Data Accessibility

An SLA is an official agreement between a physical network operator and a slice’s tenant, based on which the level of rendered service is precisely defined. It involves all relevant aspects such as performance, availability, and responsibility. In this regard, practical metrics are invoked to quantify the agreed terms. This is achieved by imposing thresholds on some KPIs including the number of assigned resource blocks, minimum throughput, backhaul capacity per slice, and so on. Furthermore, upper bounds may be defined for the global SLA violation rate to account for SLA breaching and the corresponding penalties. In practice, these SLAs might be implemented as non-convex non-differential constraints incorporated into the machine learning objective functions using the Lagrangian approach. This results in computationally prohibitive ML optimization, especially when dealing with DNN models that are optimized over large datasets. A successful SLA implementation would consist of smooth constraints that guarantee low-complexity ML processing.

ADVANCED BIG DATA ANALYTICS FOR
NETWORK SLICING

In this section, new big-data-based building blocks enabling intelligent network slicing management are proposed. This includes a low-complexity traffic predictor, resource provisioning models, and integrated SLA enforcement. The justification of the need for large datasets for training is also provided.

Low-Complexity Traffic Prediction

A key component in network slicing management is slice-level traffic prediction. Indeed, most traffic predictors have been designed to operate on slices with a large number of OTT applications,
Resource Allocation

Another building block for intelligent slicing management consists of the implementation of resource provisioning solutions. Based on the predicted traffic, our proposed scheme estimates the necessary resources per slice at each virtual network function, as depicted in Fig. 3. With this intent, DNNs are adopted to build accurate resource allocation models and train them using big datasets in an offline phase. Each sample of the datasets consists mainly of the slices’ individual traffic and the corresponding resource consumption. Using the online predicted traffic, our developed models can then concisely yield the required amount of resources in the next time step. To have an idea about the size of the datasets, let us assume that the DNN models are respecting the conditions of [14, Theorem 1]. For accuracy ε, the number of batches should be \( N_b \geq \alpha \log(b/\epsilon) \), where \( a \) and \( b \) are constants depending on the DNN depth and initialization, respectively. This translates into using very large datasets as long as one seeks small errors. To further accelerate the training and operation of the DNN resource provisioning models, one proposes that multiple instances be created for each virtual network function, and with the aid of Hadoop and/or Spark technologies, the DNN processing can be distributed and parallelized, but at the expense of extra computing resources at the different data centers.

Numerical Results

To exemplify the slicing approach proposed in this section, this article invokes a large hourly dataset stemming from a live cellular network. This dataset features the OTTs’ traffic as well as the consumed RRC connected users’ licenses per virtual baseband unit (vBBU) recorded over five days for sites located in a dense urban area. This translates into a file size of 46 MB, which is difficult to process using traditional tools such as Excel and the pandas library. Therefore, this work considers Spark technology to handle these big data. To that end, the OTT traffic is first aggregated to yield the traffic per slice. For example, the traffic of Youtube, Netflix, and Facebook video are gathered to form the enhanced mobile broadband (eMBB) slice traffic, which is then fed to the light GRU predictor detailed in Fig. 2 to yield the slice’s traffic in the next hour. As shown in Figs. 4 and 5, this simplified GRU architecture presents a reduced training/test time while ensuring accurate prediction. Next, by feeding the SLA-constrained DNN in Fig. 3 with the predicted traffic per slice, the required remote radio control

![Figure 3: SLA-constrained DNN for dynamic slicing resource provisioning.](image-url)
(RRC)-connected users’ licenses are obtained. As depicted in Fig. 6, when enforcing the SLA, the number of allocated RRC connected users’ licenses at the vBBU is respecting the agreed upper and lower bounds for the considered three slices, namely, enhanced mobile broadband (eMBB), social media, and browsing, while also following the traffic change. This is justified by the trade-off implied by the DNN optimization, which consists of a two-player game between the player minimizing the mean square error and the one achieving the SLA constraints.

CHALLENGES AND RESEARCH DIRECTIONS

While we notice great advances in the area of big-data-enabled network slicing as outlined throughout this article, we believe that there are still many challenges and open research directions that need to be addressed by the community.

INTEGRATING BIG DATA AND FAST ORCHESTRATION

In network slicing, there are many time-sensitive applications that need real-time or near-real-time analytics. Such applications call for new analytic frameworks that support big data in conjunction with fast machine learning. In this regard, large dataset reduction techniques such as principal component analysis (PCA) may accelerate the training and operation of machine-learning-based orchestration schemes. Moreover, leveraging offline-trained resource provisioning models may yield fast online allocation. Finally, pushing learning algorithms to the network edge may further reduce slices’ latency.

SECURE ORCHESTRATION

Data-driven machine learning approaches (e.g., deep learning) can be attacked by false data injection (FDI), which compromises the validity and trustworthiness of the orchestration. Resilience against such attacks is a must for ML algorithms. In this respect, joint deep learning algorithms can be developed to perform both slice resource orchestration and FDI attack detection. To guarantee fast operation, deep learning algorithms can be trained offline, while joint detection and orchestration operate in online mode. This is also in line with the aforementioned fast orchestration requirement.

FEDERATED NETWORK SLICING

This is an interoperator capability in which two operators can make network slices available on each other’s infrastructure, which would make it possible for enterprises to offer global services and still maintain consistent network reliability and quality of service. Using NFV, this technology lets workloads from a slice’s tenant be run on several physical networks. In this case, training and operating network orchestration models are challenging, given that the big datasets are distributed across many networks and would require federated learning approaches to efficiently use them as well as to enable fast orchestration.

TRANSFER LEARNING FOR NETWORK SLICING

To significantly reduce the training time of deep-learning-based slices’ resource provisioning, one may invoke transfer learning, where what has been learned for a slice setting — such as the weight of the DNN models — can be exploited to improve the generalization in another slice. This can be achieved by, for example, initializing the target slice DNN model using the weights of the source slice model. Therefore, transfer learning for network slicing is an open research area, where big data in conjunction with deep learning can be efficiently leveraged by intelligently reusing the trained models over the different slices.

CONCLUSION

To deploy network slicing, efficient resource provisioning schemes should be adopted. These schemes operate on the big data collected by the OSS to train machine learning models for slices’ traffic prediction and resource allocation. Given the need for low-complexity network management, this work has proposed a light gated recurrent unit architecture for traffic forecast. Using dataset-based constrained deep learning, the article has introduced an approach to enforce SLA constraints into the slices’ resource provisioning models. As a perspective, enabling fast learning requires the adoption of advanced big data storage and parallel processing technologies as well as features...
reduction techniques. Immunity against security attacks and federated network slicing are also open research areas for big-data-enabled network slicing.

ACKNOWLEDGMENTS

This work has been supported in part by the research projects AGAUR (2017-SGR-891), SPOT5G (TEC2017-87456-P), and 5G-SOLUTIONS (856691).

REFERENCES

[1] J. Manyika et al., “Big Data: The Next Frontier for Innovation, Competition, and Productivity,” White Paper, McKinsey Global Institute, May 2011.

[2] NGMN Alliance, “Description of Network Slicing Concept”; https://www.ngmn.org, accessed June 2019.

[3] C. Marquez et al., “How Should I Slice My Network? A MultiService Empirical Evaluation of Resource Sharing Efficiency,” Proc. MobiCom 2018, 2018, pp. 77–84.

[4] K. Cho et al., “Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation,” Proc. EMNLP 2016, Doha, Qatar, Oct. 2016, pp. 1724–34.

[5] K. Aziz, D. Zaidouni, and M. Bellafihi, “Real-Time Data Analysis Using Spark and Hadoop,” Proc. Int’l Conf. Optimization and Applications, Mohammadia, Apr. 2018, pp. 1–6.

[6] D. Bega et al., “DeepCog: Cognitive Network Management in Sliced 5G Networks With Deep Learning,” Proc. IEEE INFOCOM 2019, Paris, France, May 2019.

[7] K. He and J. Sun, “Convolutional Neural Networks at Constrained Time Cost,” https://arxiv.org/pdf/1412.1710.pdf, Dec. 2014.

[8] V. Sciancalepore et al., “Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization,” Proc. IEEE INFOCOM 2017, Atlanta, GA, May 2017.

[9] N. Laptev et al., “Time-Series Extreme Event Forecasting With Neural Networks at Uber,” Proc. NIPS Time Series Workshop, Dec. 2017.

[10] C. Zhang and P. Patras, “Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks,” Proc. MobiHoc 2018, 2018, pp. 231–40.

[11] V.-G. Nguyen et al., “SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey” Proc. IEEE Commun. Surveys Tutorials, vol. 19, no. 3, 3rd qtr. 2017, pp. 1567–1602.

[12] W. Anani, “Recurrence Neural Network Architectures Toward Intrusion Detection,” 2018, Electronic Thesis and Dissertation Repository, 5625; https://ir.lib.uwo.ca/etd/5625; accessed July 2019.

[13] S. Arora et al., “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning,” arXiv.org/abs/1811.03378, Nov. 2018, accessed: July 2019.

[14] S. Arora et al., “A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks,” Proc. ICML 2019, May 2019.

[15] A. Cotter et al., “Training Well-Generalizing Classifiers for Fairness Metrics and Other Data-Dependent Constraints,” arXiv.org/abs/1807.00028.

BIOGRAPHIES

Hatim CHergui [M’12] received his Bachelor’s degree in telecommunications engineering from the Institut National des Postes et Télécommunications (INPT), Rabat, Morocco, in 2007, and his Ph.D. degree (with honors) in electrical engineering and telecommunications from Télécom-Bretagne, Brest, France, in 2015. Since 2008, he has been a radio network planning and optimization engineer, with extensive industry experience in providing 3G/4G consulting services to Huawei Technologies, Morocco. He has also served as a radio technologies expert at the Moroccan operator INWI, Casablanca. He is the recipient of the Best Paper Award at the 2020 IEEE International Conference on Communications. He is currently a postdoctoral researcher at the Catalan Telecommunications Technology Center (CTTC) in Barcelona, Spain. His research interests lie in the area of performance analysis and machine learning applied to wireless communications.

Christos Verikoukis [SM’07] received his Ph.D. degree from the Technical University of Catalonia, Barcelona, Spain, in 2000, in the area of broadband indoor wireless communications. He is currently a Fellow Researcher with the Catalan Telecommunications Technology Center (CTTC) in Barcelona, Spain. He is the recipient of the 2011 and 2020 IEEE International Conference on Communications Best Paper Award at the 2020 IEEE International Conference on Communications. He is currently the ComSoc EMEA director, a ComSoc BoG member, and a Member-at-Large of the GITC.