A Resource Sharing Method for Reliable Slice as a Service Provisioning in 5G Metro Networks

Citation for the original published paper (version of record):
Amato, E., Tonini, F., Raffaelli, C. et al (2021)
A Resource Sharing Method for Reliable Slice as a Service Provisioning in 5G Metro Networks
2021 25th Conference On Optical Network Design And Modelling, ONDM 2021: 1-3

Network Slicing Automation: Challenges and Benefits

Federico Tonini[†], Carlos Natalino[†], Marija Furdek[†], Carla Raffaelli[*], Paolo Monti[†]
[†]Department of Electrical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden
E-mail: {tonini,carlos.natalino,furdek,mpaolo}@chalmers.se
[*]DEI, University of Bologna, 40136 Bologna, Italy
E-mail: carla.raffaelli@unibo.it

***Abstract*—Network slicing is a technique widely used in 5G networks where multiple logical networks (i.e., slices) run over a single shared physical infrastructure. Each slice may realize one or multiple services, whose specific requirements are negotiated beforehand and regulated through Service Level Agreements (SLAs). In Beyond 5G (B5G) networks it is envisioned that slices should be created, deployed, and managed in an automated fashion (i.e., without human intervention) irrespective of the technological and administrative domains over which a slice may span. Achieving this vision requires a combination of novel physical layer technologies, artificial intelligence tools, standard interfaces, network function virtualization, and software-defined networking principles. This paper provides an overview of the challenges facing network slicing automation with a focus on transport networks. Results from a selected group of use cases show the benefits of applying conventional optimization tools and machine-learning-based techniques while addressing some slicing design and provisioning problems.**

***Index Terms*—Network slicing, Automation, Machine learning, Beyond 5G**

I. Introduction

5G is entering the early deployment phase. It will support an unprecedented number of services (e.g., enhanced Mobile Broadband (eMBB), massive Machine Type Communication (mMTC), and Ultra Reliable and Low Latency Communication (URLLC)). Network slicing allows the provisioning of these different services over a common physical infrastructure. It creates end-to-end logical networks (i.e., the slices) by assigning virtual and/or physical resources to different slices with the guarantee that the performance requirements of specific service are met [1].

An example of a typical architecture leveraging on the slicing concept is presented in Fig. 1. It comprises several technological and/or administrative domains in a hierarchical manner [2]. Data plane resources are controlled and monitored by domain-specific controllers (i.e., one for each domain). On top of them, one or more orchestration layers act to provide multi-domain, end-to-end services. Once a service provider requests a slice, the orchestration layer decides if the request can be satisfied by the network. If a slice is admitted, a proper set of resources has to be assigned to meet the Service Level Agreements (SLAs). While the network is in operation, telemetry information is gathered and exchanged between the data and the control plane to monitor the state of resources and services currently running in the network. In this process, domain-specific information must be abstracted and exchanged through standard or proprietary communication protocols. Thanks to this data, the control and orchestration layers (equipped with the right tools) are then able to adapt to traffic changes (e.g., scale up or down slices), to detect potential faults/security breaches, and to take proper countermeasures. The entire process needs to be reliable, secure and autonomous, to reduce as much as possible human intervention. This, in turn, is one of the cornerstones of Beyond 5G (B5G) networks that aim at creating automated and trustworthy network environments [3].

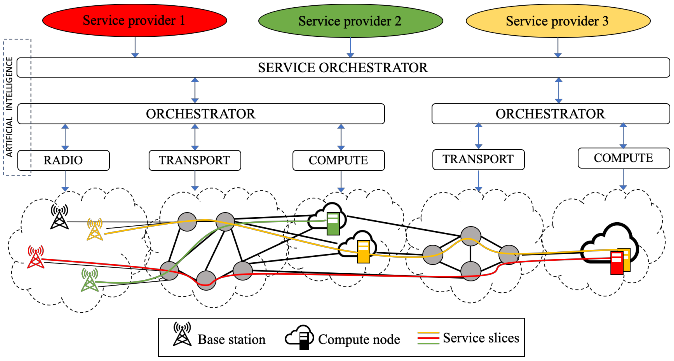


Fig. 1: Example of a hierarchical architecture for 5G network slicing.

In this paper, we highlight the progress and main challenges in achieving a fully automated slice deployment. While the network slicing concept can be applied to all technological domains, we primarily focus on the transport network resources. Selected results show how different algorithms based on machine learning can be used in slice admission control and attack detection, and how different levels of reliability impact the backup resources to be provisioned in a URLLC scenario.

II. State of the art and challenges

The section summarizes the state of the art on network slicing. Each subsection focuses on a specific topic highlighting what it has been accomplished and what are still the open questions that need to be addressed.

This study was financed in part by the project "Smart city concepts in Curitiba low-carbon transport and mobility in a digital society" sponsored by VINNOVA.

A. Transport Network Slicing and Technologies

When looking at transport networks, slicing techniques should support services with different requirements (e.g., in terms of data rates and delay) while differentiating the traffic flowing in and the resources used by each slice (i.e., isolation). For example, Virtual Local Area Networks (VLANs) or Multiprotocol Label Switching (MPLS) can be used to tag traffic from different users through labels or IDs or using other forms of encapsulation [4]. Depending on the technology used by a specific data plane, other technologies, such as Flexible Ethernet (FlexE), Optical Transport Network (OTN), and time/wavelength division multiplexing can be used to provide stronger isolation while guaranteeing a specific amount of physical resources for each service [5].

Transport network resources can be sliced in different ways depending on the use case. For example, a mobile network provider may need different connectivity services depending on which radio splits are used [6]. Physical layer splits may need hundreds of Gbps of capacity, depending on the antenna configuration. This, in turn, requires a very high capacity transport system. Space Division Multiplexing (SDM) techniques can help in this respect allowing to carry high capacity fronthaul traffic. Spatial different SDM resources (e.g., cores and modes) can be assigned to different services, enabling isolation [7].

Splits can be also changed at run-time, depending on the network conditions (e.g., interference) and resource availability [8]. The adoption of different splits brings strict latency requirements. Some are delay-tolerant, while other split options are time-critical, calling for time-sensitive and deterministic transport solutions that can adapt to evolving traffic conditions. As a result, IEEE recently formed the Time-Sensitive Networking (TSN) working group intending to devise solution for carrying high and low priority traffic together. They propose to introduce deterministic delay via time synchronization in Ethernet networks [5]. Conversely, fusion is a different option effective in inserting best-effort traffic in real-time streams by adding fixed and bounded delay (and no jitter) to high priority streams, with no need for time synchronization [9]. Slicing can be provided with these architectural solutions but studies on efficient traffic scheduling are still missing.

B. Attacks, Security and Vulnerabilities

A network infrastructure can be attacked to steal sensitive information and/or to disrupt traffic. This, in turn, requires encryption, authentication, and integrity check mechanisms to be put in place. Threats to physical resource, virtual functions, and software platforms must be quickly discovered and actions must be taken promptly to avoid service outages [10]. ML-based techniques can help in finding relationships among heterogeneous data, where explicit models or complete information are not available. However, this requires (i) integration of ML modules with existing platforms, also offering support to legacy and current-generation devices, and (ii) proper ML models that match the specific use case and address the inaccuracies due to false positives/negatives.

Slice isolation techniques prevent different services to potentially depleting slice resources, or to exhaust common resources with multiple slices, causing Denial of Service (DoS) to other subscribers. Distributed DoS attacks may also be caused by malware on user's devices, and since they may be connected to different slices simultaneously, this could lead to unwanted inter-slice communication [11]. Isolation is also required to avoid resources assigned to a slice to be accessed by other slices, especially for privacy reasons (e.g., personal data stored in a data center). For example, if a network function (NF) is shared, a violation of the NF may allow attackers to steal information from different slices [10]. Isolation of NFs can be done, e.g., at the hardware level, the virtual machine, or kernel level [12]. Complete isolation NF is preferable from a security point of view. However, this usually leads to different dedicated networks with very low multiplexing gains. Since slices with different security requirements must be provided on the same infrastructure, additional studies are required to investigate how to provide the proper level of isolation, depending on the specific service.

Network virtualization and softwarization introduce also several vulnerabilities to attacks [13] [14]. The separation of the control and the data plane exposes the network to potential attacks. Management interfaces between network entities must be secured to avoid impersonation of slice managers, which may lead to theft of sensitive information, creation/termination of slice instances and other unauthorized activities. A breach of data plane functions could also result in a control plane violation [11]. Even though sensitive data can be encrypted, side channels attacks, where an attacker collects information that is usually exchanged in the clear (e.g., metadata), can be conducted. These data could be used, for example, to induce faults or tamper with the system cache [11]. All these aspects must be studied in the context of 5G and B5G frameworks. The delay and computational effort introduced by different levels of encryption may impact the slice design and require additional studies. The most appropriate level of security countermeasures must be investigated, depending on the service type.

Transport network slices may also be affected by physical layer attacks, e.g., via signal jamming or external polarization modulation. In- or out-of-band jamming can significantly degrade signals, both wireless or wired, or may target control channels, resulting in a denial of access for selected users [15], [16]. Polarization modulation attacks in fibers induce demultiplexing errors [16]. Techniques to detect and mitigate the errors induced by these attacks must be provided. Different ML-based techniques can be employed to detect and classify attacks or find anomalies in optical networks, requiring access to data at different control and orchestration levels [17]. Approaches based on hierarchical learning can be applied in multi-domain scenarios to hide domain-specific information [18]. However, this requires to study accurate abstraction policies to avoid the effects of error propagation while keeping reasonable scalability performance. In addition, ML models also present inherent inaccuracy, especially when new data are introduced. To compensate, error mitigation techniques can be considered, e.g., by combining multiple ML models. This requires finding a trade-off among model complexity, accuracy, and time needed to compute the solution (including training).

C. Optimal Resource Allocation Strategies

Upon acceptance, a slice needs to be mapped into the network infrastructure. In turn, this becomes an optimization problem where only the right amount of network resources should be provided while guaranteeing the performance level required by a specific service. Otherwise, overprovisioning of resources might lead to an increased slice rejection rate or to service degradation, with an obvious impact on the revenue of the service/infrastructure provider.

Slice resource mapping can be seen as an extension to the well studied Virtual Network Embedding (VNE) problem [19]. More specifically, VNE finds (i) the optimal placement of NF and (2) the best allocation of the virtual link capacity required to interconnect the NFs. The solution to a slice resource mapping problem might also have to identify the most appropriate data plane technology to be used as well as a proper set of backup resources (local vs. end-to-end, dedicated vs. shared) to support the required level of resiliency in the presence of failures [20]–[22]. In some cases, only specific functions or paths within a single slice need to be protected, requiring additional resources to be provided as a backup. In other cases, with extreme reliability requirements, backup resources must be provisioned in a 1+1 or 1:1 manner. When the problem becomes very complex, game-theoretic approaches can be used to consider the relationship among users, network operators, and service providers, to formulate optimization problems. Examples are the fairness of network resource allocation, profit maximization, and cost minimization of network slice's users [23].

It should be noted that the slice resources requirement profile may change over time (e.g., users might behave differently at a different time of day) and a mere peak-based resource assignment could result in low service acceptance. This calls for strategies aimed at reconfiguring slices over time. As a result, there are approaches in the literature that support the scaling up/down of slices at run-time based on the level of utilization of the network resources [24]. Another example of slice adaption is the possibility to vary the choice of baseband splits over time, while the slice is in operation. This approach helps to achieve better resource utilization and to reduce the transport network load. However, this may require frequent reconfigurations to pursue cost minimization and therefore trade-offs must be derived [25]. Prediction models can be also be adopted to forecast network changes and take actions on the network resources in advance. These aspects are analyzed in the next section.

D. Slice Management and Orchestration

In the most general case slices might span across multiple administrative domains while combining resources belonging to different technological domains. This calls for the development of a standard way to exchange information and interactions among different providers and domains, to ensure that SLAs are met. Also, an appropriate set of data must be selected and sent to the orchestration layer to (i) solve the slice admission and mapping problems in an optimal way and (ii) to be able to continuously monitor the SLAs during the slice lifecycle.

Establishing a multi-domain slice leverages on the principle of recursive virtualization and hierarchical network abstraction [26]. The network resources allocated to a particular tenant can be abstracted and exposed to a third party that can construct a new service on top of the prior one. This approach simplifies the composition of slices allowing a combination of different resources in a flexible way. Upon the arrival of a slice request, the service orchestration layer decides whether to admit the slice or not. This process involves the identification of the domains to be involved. Then, the slice request must be converted into directives for the different domains, that must select the most appropriate set of resources. This can be done using an intent-based networking paradigm, which allows expressing slice requirements and constraints in the form of policies [27]. Each domain is also responsible for providing monitoring data throughout the slice life-cycle. Data from different domains are collected and elaborated by the service orchestration layer to monitor SLAs and take the necessary actions. The interactions among these entities can be based on a peer to peer approach or a federated infrastructure domain [2]. In the former, orchestrators of different domains interact to find a solution that satisfies specific SLA. In the latter, a common cross-domain slice coordinator leverages trusted connectivity across administrative domains and carries out domain-specific resource allocation.

In terms of challenges, end-to-end management and orchestration frameworks require the implementation of specific functionalities to reach complete automation. Slice deployment should be autonomous, requiring automatic acceptance or denial of slice requests, based on the network resources and service requirements. The network control should also be able to continuously monitor the state of the resources to adapt the slice mapping to the evolving network conditions, i.e., to be able to re-configure itself. This is required, for example in the case of traffic variarion and/failures. All this must work when slices traverse different technological and/or administrative domains, requiring to elaborate and expose information among the different entities in a common, standard way.

Artificial intelligence (AI) allows the creation of systems that autonomously take decisions based on their perceived environment. To do so, information must be collected from the network, where equipment of different suppliers co-exists, requiring the definition of common standard interfaces to create vendor-agnostic monitoring systems [28]. Moreover, telemetry information can be exploited for proactive or reactive network re-configurations. Different models can be used to obtain information about the physical layer and trigger changes at the network level, e.g., in routing, spectrum and modulation assignments [29]. These data can also be used to estimate the traffic and take appropriate actions [30], i.e., triggering reconfiguration strategies to change the current slice resource assignment and avoid SLA violation or slice request rejection

[28] [31]. Even though these approaches are effective, further analysis of the computational effort and performance of these strategies, as well as the amount of data to be collected and elaborated in real-size scenarios require further studies.

Finally, from the standardization point of view efforts are needed to enhance current information models to account for multi-domain connectivity and control, resiliency and performance measurements, as well as multi-domain intent-based networking interfaces. From a resource abstraction point of view, the functions and connectivity resources to be exposed impact the end-to-end network performance and cost. Finally, security aspects are yet to be considered in multi-domain scenarios, e.g., how to guarantee authorization, integrity, and encryption among different players.

III. Benefits of Automation: a few results

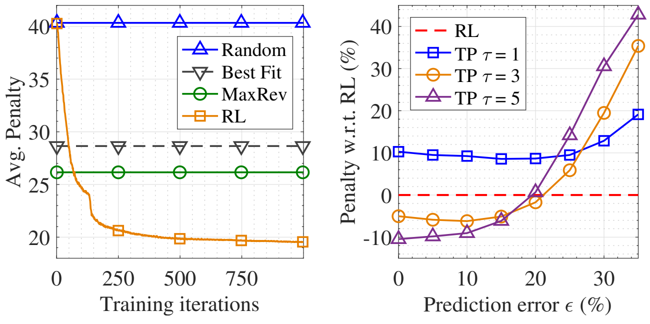
This section reports a number of selected results while addressing some of the challenges described above. The use cases under exam include: slice admission control, optimal URLLC slice deployment strategies, and application of ML-based method to address security problems.

A. *Intelligent Slice Admission Control*

Different strategies based on ML can be employed in the slice admission process. For example, reinforcement learning (RL) strategies can be used to make scheduling decisions based on the feedback derived from past actions. Another possibility is to use supervised learning methods to derive traffic predictions (TP) to be used to get insight into future resource needs.

The work in [31] reports a comparison of the two approaches. In RL, the inputs for the neural network are the value of the service holding time, the number of required resources and the current status of the infrastructure. The output of the neural network indicates the best service in the queue to be provisioned. The reward function is proportional to the sum of the penalties associated with the services currently waiting in the buffer and to the ones currently being provisioned in the infrastructure. At each training iteration, to minimize the value of the reward function, the discounted reward is computed and a policy network is optimized using the gradient descent method. The TP-based heuristic provisions a service with the help of a regression-based TP function that estimates the resources required by each one of the services included within the prediction window τ. A TP-based heuristic checks each service in the queue and selects the service for which (i) enough resources are available, and (ii) its provisioning generates the lowest penalty as compared to selecting other services in the buffer. Each service expected to be requested within τ is provisioned if the penalty incurred by the services within τ when the service is provisioned is greater than the same penalty when the service is not provisioned. Otherwise, the service is held in the queue.

Two different kinds of services are considered. A mobile service provider (MSP) requires the activation of up to 3 small cells, each requiring a dedicated wavelength, and a total service capacity of up to 4 CPUs in a data center, for a duration of up to 3 time steps. A cloud service provider (CSP) requires services with up to 2 wavelengths and 4 CPUs each, for a duration of $[10, 15]$ time steps. All these variables are randomly selected using independent uniform distributions. 80% of all the services are MSP, while 20% are CSP. The infrastructure provider pays a penalty proportional to the delay (i.e., measured in time steps) in provisioning a service. The penalty coefficient of the MSP services is five times bigger than the one of the CSP services. More details are available in [31].



(a) (b)

Fig. 2: Comparison of average penalty for RL, Random, Best Fit, and MaxRev (a), and RL vs. TP for different prediction windows τ and prediction errors ϵ (b).

Fig. 2a shows how the RL is able to reduce the penalty factor with respect to three benchmarks: (i) Random, that selects which service to provision with a uniform probability, (ii) Best Fit, that selects the service that fits best the available resources, and (iii) Maximum Revenue (MaxRev), that prioritizes MSP services over the CSP ones. In the beginning, the RL performs similarly to the Random strategy. When the number of iterations increases, RL learns that it is more beneficial to serve MSP over CSP services to keep the penalty factor low. Fig. 2b compares the performance of RL with the TP-based heuristic for different prediction windows. It shows that when $\tau = 1$, the performance of RL is always better regardless of the value of the prediction error ϵ. For low values of ϵ, larger prediction windows allow TP to outperform RL by up to 6% and 10% in case of $\tau = 3$ and 5, respectively. However, if ϵ is large (i.e., $> 20\%$), there is no gain in using a TP-based heuristic.

B. *Optimal URLLC Service Slice Deployment*

Network failures impact the operation with consequences on the service provisioning. Among all the services envisioned for 5G, URLLC are the most critical ones. The deployment of a URLLC service slice requires provisioning of radio, transport and cloud resources. In addition, provisioning of additional backup resources must be considered, to be used in case of failures. Backup resources can be dedicated or shared, depending on the specific requirement, whereas in the latter case some time is needed to switch to the backup resources when a failure is detected.

TABLE I: Active nodes and capacity savings for 6 node network in the balanced and unbalanced cases under 2 and 3 hop constraints. The unconstrained case relaxes limitations on the network resources and delay.

	Active nodes		Saved
Network	DPP	SPP	Capacity
2 hops - bal	6	4	66.6%
3 hops - bal	6	4	66.6%
2 hops - unbal	4	4	23.6%
3 hops - unbal	4	4	27.8%
Unconstrained	2	2	0%

Two different ILP models are presented in [22] comparing the outcome of dedicated and shared backup path protection (DPP and SPP, respectively). Both strategies allow to select the best baseband split depending on the available network resources while providing resiliency against a single node or link failure. The objective of these strategies is to minimize the number of nodes where to install cloud resources and the amount of resources to be provisioned. Numerical results are obtained considering the deployment of a URLLC slice in a 6 node network, under different conditions. Two different resource distributions are considered. In the balanced case, all the nodes have the same capacity (25 processing units - PUs). In the unbalanced case, two nodes are assumed to have unlimited resources, while other nodes have limited capacity (10 PUs). All the links have the same and limited capacity (40 Gbps). More details are available in [22].

Table I shows the number of nodes selected to host either baseband, core or cloud functions (referred to as active nodes) and the saved backup computational capacity when SPP is used, with respect to the DPP, in the sample network. The unconstrained case, reported as a benchmark, provides a lower bound for the number of active nodes, that is the case when no constraints on capacity, bandwidth, and latency are applied. Since this case requires only 2 nodes, it also exhibits no backup resource sharing. In real case scenarios, when resources are limited, the number of nodes increases due to finite node and link capacity. This situation is evaluated under two different delay constraints (2 and 3 hops). In the balanced case, sharing backup resources leads to a reduction in the number of nodes to be activated, regardless of the number of hops. This is due to the sharing of backup resources in both links and nodes, that allows reducing the backup capacity by 66.6%. In the unbalanced case instead, where some nodes provide extensive capacity, the SPP is still effective in sharing backup capacity with a reduction of up to 27.8%.

C. Machine Learning in Optical Network Security

Attacks targeting the physical layer of optical networks can cause service outages involving one or more slices, depending on the level of resource sharing and isolation. Different attacks can cause optical parameters to deviate from regular operating conditions. Existing models of physical layer impairments are too simplistic to capture the complex effects of a range of attacks [17]. Instead, ML techniques have found a useful application in identifying intricate patterns among different parameters. Supervised, semi-supervised, and unsupervised learning techniques can be used to identify security breaches by jointly analyzing multiple monitoring parameters. Supervised learning models can be trained to learn the trends in optical performance indicators that characterize different attacks and normal working conditions, potentially providing fine-granular classification of the attack type and intensity, depending on the training set. However, it is not easy to provide a representative and precise set of correctly labeled data for a continuously evolving attack landscape. Under such circumstances, semi-supervised learning models can be used for detecting the presence of an attack even if it is previously unseen, but without the ability to categorize it. This is obtained through training using only the data from normal operating conditions. Unsupervised learning models can be used to detect attack presence in the network in circumstances when no training data is available. These three techniques exhibit different requirements in terms of data acquisition before their use, in addition to different levels of performance obtained during their use. Depending on the considered scenario, the most appropriate technique to be used needs to be carefully evaluated.

In the following, we consider a real test-bed subject to physical-layer attacks described in [16]. Three different attacks have been considered: in- and out-of-band jamming, and external polarization modulation. The first two attacks consist in inserting harmful signals generated by a continuous wave laser into a breached fiber, propagating in the same or in an adjacent optical channel as the channel under test. Polarization modulation, instead, is performed by squeezing the fiber with a modulator driven by a sine-wave generator, inducing changes in the state of polarization that are too fast for the coherent receiver to compensate for, which results in erroneous detection. Twelve different Optical Performance Monitoring (OPM) parameters are collected from the network. Three different models have been used for attack detection: (i) a supervised learning model using Artificial Neural Network (ANN), (ii) a semi-supervised learning model using One-Class Support Vector Machine (OCSVM), and (iii) an unsupervised learning model using Density-Based Spatial Clustering of Applications (DBSCAN). The results are summarized in Table II, where the false positive and negative rates, and the $f1$ score are reported for each model. The ANN is always capable of detecting the attacks although sometimes it might classify them in the wrong attack category. In the semi-supervised and unsupervised models, instead, there is a probability that an attack will remain undetected (false negative) and a probability that a normal operating condition is flagged as an attack (false positive). The $f1$ score is a measure that considers both false positive and negative rates providing a unified accuracy metric. In the case of OCSVM and DBSCAN, the false positive rate is 0.029 and 0.062, respectively, while the false negative rate is 0.003 and 0, respectively. In case of ANN, the $f1$ score is 1, as it is able to detect all attacks, while in case of OCSVM and DBSCAN it is lower. In particular, the OCSVM provides very good performance with a score of 0.985, while for the DBSCAN the score is 0.970. These results indicate

TABLE II: Comparison of the performance for different ML models showing false positive and negative rate, and $f1$ score.

ML model	False positive rate	False negative rate	$f1$ score
ANN	0	0	1
OCSVM	0.029	0.003	0.985
DBSCAN	0.062	0	0.970

that, in the considered scenario, the lower the data acquisition overhead, the larger is the error provided by the solution. Depending on the specific case, techniques aimed at improving the performance of unsupervised and semi-supervised learning can be adopted. For example, a time window based approach allows to trigger countermeasures if the attack is detected in several of consecutive samples, which reduces the probability of false alarms or attack misdetection over the window at the expense of introducing some delay in the attack detection process.

IV. Conclusions

This paper presents an overview of the work currently underway to address a number of challenges in the area of transport network slicing. The paper also elaborate on the main challenges to that needs to be addressed to reach a fully automated slice deployment scenario, one of the key features for beyond 5G networks. Results on a number of selected use cases show how different algorithms based on machine learning can be used in slice admission control and attack detection, and how different levels of reliability impact the backup resources to be provisioned in a URLLC scenario. The discussion of the main challenges suggests that several aspects remain unexplored. Resource assignment and re-allocation techniques in dynamic scenarios must be studied, especially with novel equipment supporting time-sensitive networking. In addition, multi-domain orchestration frameworks still lack a standardized way to communicate with other domain orchestrators. Proper data sets to be exchanged in this view (e.g., monitoring parameters or abstracted resources) as well as the impact on the network performance require additional investigation. Also, artificial intelligence is indeed a powerful tool towards secure and autonomous networks, but still lacks a deeper scalability analysis.

References

[1] "Description of network slice concept, v1.0," NGMN Alliance, Jan. 2016.

[2] T. Taleb *et al.*, "On multi-domain network slicing orchestration architecture and federated resource control," *IEEE Network*, vol. 33, no. 5, pp. 242–252, Sep. 2019.

[3] "New services and capabilities for network 2030: Description, technical gap and performance target analysis," ITU-T FG NET-2030 Sub-G2, Oct. 2019.

[4] X. Costa-Pérez *et al.*, *Network Slicing for 5G Networks.* John Wiley and Sons, Ltd, 2018, ch. 9, pp. 327–370. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119333142.ch9

[5] P. Sehier *et al.*, "Transport evolution for the RAN of the future [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B97–B108, April 2019.

[6] R. Ferrus *et al.*, "On 5G radio access network slicing: Radio interface protocol features and configuration," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 184–192, May 2018.

[7] S. G. Leon-Saval, N. K. Fontaine, and R. Amezcua-Correa", "Photonic lantern as mode multiplexer for multimode optical communications," *Optical Fiber Technology*, vol. 35, pp. 46 – 55, 2017, next Generation Multiplexing Schemes in Fiber-based Systems.

[8] Y. Li *et al.*, "Flexible ran: Combining dynamic baseband split selection and reconfigurable optical transport to optimize ran performance," *IEEE Network Magazine*, 2020.

[9] S. Bjornstad *et al.*, "Minimizing delay and packet delay variation in switched 5G transport networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B49–B59, 2019.

[10] "The evolution of security in 5G," 5G Americas, Jul. 2019.

[11] V. A. Cunha *et al.*, "Network slicing security: Challenges and directions," *Internet Technology Letters*, vol. 2, no. 5, p. e125, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.125

[12] Z. Kotulski *et al.*, "On end-to-end approach for slice isolation in 5G networks. fundamental challenges," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2017, pp. 783–792.

[13] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 623–654, Firstquarter 2016.

[14] M. Pattaranantakul *et al.*, "NFV security survey: From use case driven threat analysis to state-of-the-art countermeasures," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3330–3368, Q4 2018.

[15] M. Lichtman *et al.*, "5G NR jamming, spoofing, and sniffing: Threat assessment and mitigation," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[16] C. Natalino *et al.*, "Experimental study of machine-learning-based detection and identification of physical-layer attacks in optical networks," *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4173–4182, 2019.

[17] M. Furdek and C. Natalino, "Machine learning for optical network security management," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, 2020, pp. 1–3.

[18] G. Liu *et al.*, "Hierarchical learning for cognitive end-to-end service provisioning in multi-domain autonomous optical networks," *Journal of Lightwave Technology*, vol. 37, no. 1, pp. 218–225, 2019.

[19] S. Vassilaras *et al.*, "The algorithmic aspects of network slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, Aug 2017.

[20] N. Shahriar *et al.*, "Reliable slicing of 5G transport networks with dedicated protection," *CoRR*, vol. abs/1906.10265, 2019. [Online]. Available: http://arxiv.org/abs/1906.10265

[21] M. Lashgari *et al.*, "Cost benefits of centralizing service processing in 5G network infrastructures," in *Asia Communications and Photonics Conference (ACPC) 2019.* Optical Society of America, 2019, p. M3C.2.

[22] F. Tonini, E. Amato, and C. Raffaelli, "Optimization of optical aggregation network for 5G URLLC service," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec 2019, pp. 1–6.

[23] R. Su *et al.*, "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," *IEEE Network*, vol. 33, no. 6, pp. 172–179, Nov 2019.

[24] M. R. Raza *et al.*, "Dynamic slicing approach for multi-tenant 5G transport networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 1, pp. A77–A90, 2018.

[25] G. Wang *et al.*, "Reconfiguration in network slicing—optimizing the profit and performance," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 591–605, June 2019.

[26] I. Afolabi *et al.*, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2429–2453, thirdquarter 2018.

[27] T. Subramanya, R. Riggio, and T. Rasheed, "Intent-based mobile backhauling for 5G networks," in *2016 12th International Conference on Network and Service Management (CNSM)*, Oct 2016, pp. 348–352.

[28] D. M. Gutierrez-Estevez *et al.*, "Artificial intelligence for elastic management and orchestration of 5G networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 134–141, October 2019.

[29] F. Musumeci *et al.*, "An overview on application of machine learning techniques in optical networks," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1383–1408, Secondquarter 2019.

[30] D. Rafique and L. Velasco, "Machine learning for network automation: overview, architecture, and applications [invited tutorial]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D126–D143, Oct 2018.

[31] C. Natalino *et al.*, "Machine learning aided orchestration in multi-tenant networks," in *2018 IEEE Photonics Society Summer Topical Meeting Series (SUM)*, July 2018, pp. 125–126.