Dynamic Network Slicing and Resource Allocation for 5G-and-Beyond Networks

Alaa Awad Abdellatif[1], Amr Mohamed[1], Aiman Erbad[2], and Mohsen Guizani[3]
[1] College of Engineering, Qatar University, Qatar.
[2] College of Science and Engineering, Hamad Bin Khalifa University, Qatar.
[3] Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, UAE.

2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771877

***Abstract*—5G networks are designed not only to transport data, but also to process them while supporting a vast number of services with different key Performance Indicators (KPIs). Network virtualization has emerged to enable this vision, however it calls for designing efficient computing and network resource allocation schemes to support diverse services, while jointly considering all KPIs associated with these services. Thus, this paper proposes a dynamic network slicing and resource allocation framework that aims at maintaining high-level network operational performance, while fulfilling diverse services' requirements and KPIs, e.g., availability, reliability, and data quality. Differently from the existing works, which are designed considering traditional metrics like throughput and latency, we present a novel methodology and resource allocation schemes that enable high-quality selection of radio points of access, resource allocation, and data routing from end users to the cloud. Our results depict that the proposed solutions could obtain the best trade-off between diverse services' requirements when compared to baseline approaches that consider partial network view or fair resource allocation.**

***Index Terms*—5G networks, Network Function Virtualization, Software-defined networking, RAN slicing.**

I. Introduction

One of the major objectives of 5G-and-Beyond networks is to fulfill the communication and computation requirements of various industries (or verticals) in order to run a wide range of services with varied demands. Network Function Virtualization (NFV) has emerged as a promising technology to realize next-generation services, where network functions can be easily implemented and dynamically allocated to different services [1]. Indeed, different verticals specify the requirements and specification of their services, using a graph of virtual network functions (VNFs), then network operators map these requirements into infrastructure management decisions. This calls for designing innovative schemes for optimizing radio access networks association and VNFs allocation across the network. Importantly, which and how many resources should be allocated to each service is a critical issue, since the associated cost and availability affect the performance significantly. These resources are ranging from those in the cloud to the ones at the edge of the network infrastructure (i.e., through multi-access edge computing (MEC)) or in the fog (i.e., in devices such as smartphones and robots) [2].

Several works have been proposed for network slicing in order to support different vertical applications over 5G networks [3], [4]. For instance, an integrated NFV-based management architecture for dynamic deployment of instances of virtual tenant networks has been studied in [5]. The authors in [6] aim at maximizing the throughput by jointly selecting the optimal functional split and routing path from an End-User (EU) to the central unit. Also, the authors in [7] tried to simplify the problem of VNF placement by leveraging temporal variability of the traffic demand, hence optimizing the allocated resources. In [8], reinforcement learning approach was used for predicting traffic demand, hence achieving a near-optimal placement of VNFs with minimum costs. However, most of the presented studies either considered specific scenarios, where all network management decisions are made by a centralized entity, often the NFV orchestrator, or focused only on throughput and latency as performance metrics.

Differently from the related work in the literature, our framework aims at serving, as a case study, diverse Intelligent-healthcare (I-health) services, hence KPIs. Thus, it is designed to support multiple, heterogeneous KPIs in an efficient manner, while accounting for all types of resources and their locations (from the edge to the cloud). This calls for introducing new problem formulation with novel solutions based on distributed optimization, which is also unique and provides an efficient way to trade-off between the optimality and complexity. Thus, our main contributions can be summarized as follow: (i) proposing a distributed optimization framework that tackles all network slicing KPIs for diverse services, while allowing the VNFs to be placed at any layer of the network topology; (ii) formulating and solving the inter-slice and intra-slice allocation problems using efficient, distributed solutions that jointly create and configure the end-to-end network slices while also allocating the needed resources to the attached EUs; (iii) solving the complex (NP-hard) network slicing problem with a distributed, less complex framework, leveraging the proposed multi-stage solution.

The rest of the paper is organized as follows. Section II introduces the system model and KPIs that are considered in this paper. Section III presents the formulated inter-slice and intra-slice allocation problems for optimizing the allocated resources to different slices and EUs. Section IV presents our distributed approach for solving the proposed optimization problems. Finally, Section V demonstrates our simulation results, while Section VI concluding the paper.

II. System Model and key Performance Indicators

A. Network slicing architecture and services

This paper consider as a case study a wireless heterogeneous I-health system, where diverse health-related services, such as remote monitoring applications and remote surgeries, should be assigned a customized network slice to support its stringent Quality of Service (QoS) requirements. In our model, each service $s \in S$ is described through a *service graph*, where vertices are VNFs, $v \in V$, and

edges refer to the order of the VNFs, i.e., how traffic data should be routed from a VNF instance running on a network node to the next [9]. VNFs can represent different functionalities including: data collection, event detection, feature extraction, adaptive compression, as well as database-related functionalities [10]. Such VNFs consume/reserve diverse computing and storage resources on different network nodes. Each service s will be associated with one or more KPIs, namely, expected traffic load, maximum allowed delay, minimum level of reliability, and expected cost for providing the service. We remark that not all KPIs have to be fulfilled for all services; each service may be associated with one or more of these KPIs. For instance, remote surgeries must be associated with ultra-reliable low-latency communication (URLLC), while remote monitoring applications can be associated with energy-efficient, low-cost communication. Moreover, it is assumed that each patient (or EU) can forward his/her medical data to the cloud via multiple Radio Access Networks (RANs) [11]. Each RAN will have different characteristics, such as energy consumption, monetary cost, and transmission delay, while the multiple RANs are sharing the same control plane.

The available computing and communication resources at the fog, MEC, and cloud can be represented through the *physical graph*, whose vertices are the network nodes and endpoints (i.e., the origins or destinations of service traffic), while the edges (i, j) representing the physical links between the nodes. Network nodes may be equipped with different computing capabilities (e.g., CPU or memory), hence the quantity of resources of type k available at node n is defined by $a_n(k)$. For the sake of consistency, the service flows and physical flows must be matched. Also, the VNFs are placed only at nodes where all the needed radio interface(s) are available. Each edge (i, j) refers to a specific link $l \in \mathcal{L}$, which is associated with communication delay $D_{i,j}$ and link capacity W_l. Moreover, to model real-world quality of communication links and network nodes, each node $n \in \mathcal{N}$ and link l are associated with reliability parameters $\eta_n(t)$ and $\eta_l(t)$, respectively, which present the efficiency of a node or link to work as intended at time t.

B. key performance indicators

Now, we discuss different KPI that will be considered in our model.

1) Service latency: it comprises two main components, i.e., network delay due to transferring data traffic through different network's links, and processing delay at the nodes hosting VNF instances. The average network delay is computed by summing the delays associated with different links taken by a flow f over path p:

$$d_n(f,p) = \sum_{(i,j)\in p} D_{i,j} = \sum_{(i,j)\in p} \frac{B(f,i,j)}{r_{f,i,j}} + \epsilon_{i,j}, \quad (1)$$

where $\frac{B(f,i,j)}{r_{f,i,j}}$ and $\epsilon_{i,j}$ are the transmission time and the access channel delay of flow f expects to experience when traversing link i, j, respectively, and $B(f, i, j)$ is the quantity of a traffic flow f that has been processed last at node i, and will be next processed at node j. Following the processing model in [9], the VNF instances are modeled as M/M/1-PS queues. The choice of the processor sharing (PS) model closely emulates the behavior of a multi-threaded application running on a virtual machine (without loss of generality, different processing models can be also incorporated in our framework easily). Hence, by assuming that the quantity of traffic associated with flow f and processed at the instance of VNF v at node n is $\hat{B}(f, n)$, the total processing delay incurred by flow f over path p is written as:

$$d_p(f,p) = \sum_{n\in p} \lambda(f,n) \frac{1}{a_n(k,cpu) - r_{cpu}(n)\hat{B}(f,n)}. \quad (2)$$

where $\lambda(f, n)$ is the fraction of the traffic flow $B(f, n)$ processed at the instance of VNF v located at node n, i.e., $\hat{B}(f, n) = \lambda(f, n) \cdot B(f, n)$. We remark here that the assigned CPU has a major role in (2) compared to other types of resources. Indeed, the assigned CPU gives us an additional degree of freedom to trade-off between the cost and the performance, where assigning more CPU results in shorter processing delay, but higher costs. Giving that D_f is the maximum target delay for flow f of service s, the latency constraint for path p is written as: $d_n(f, p) + d_p(f, p) \leq D_f$.

2) Service reliability and temporal availability: The reliability of a path is computed as the product between the reliability values of all links and network nodes belonging to this path. Hence, the reliability constraint of flow f traversing through path p at time t is stated as $\prod_{n\in p}\prod_{(i,j)\in p} \eta_n(t) \cdot \eta_{i,j}(t) \geq H_f$, where H_f is the maximum target reliability required for flow f.

3) Encoding distortion: Given the enormous amount of generated data traffic in I-health systems, it is typically unpractical to transfer the entire raw data from the EU to the cloud [12]. Hence, it is important to implement, at the network edge, adaptive compression techniques. This enables, on one hand, decreasing the transmitted data size, hence decreasing the transmission energy consumption. On the other hand, leveraging lossy compression techniques produces an encoding distortion. Thus, in our framework, we consider, as a case study, the encoding model of the Electroencephalogram (EEG) signals presented in [13][1]. In this model, the distortion is measured by the percentage root-mean-square difference between the recovered data and the original one. Using our real-time implementation in [13], the obtained distortion at node n for flow f is formulated as

$$\psi(f,n) = (x_1 e^{(1-\kappa(f,n))} + x_2(1-\kappa(f,n))^{-x_3} + x_4)/100, \quad (3)$$

where $\kappa(f, n)$ is the data compression ratio at node n, and $x_1 \rightarrow x_4$ are the model parameters that can be estimated using statistics of the considered EEG encoder [13].

4) Energy consumption: At EU i, the transmission energy consumption to transfer a traffic of length B_i bits over RAN j with rate r_{ij} is stated as:

$$E_{ij} = \delta_j \left(\frac{B_i \cdot N_0}{r_{ij} \cdot g_{ij}} (2^{r_{ij}} - 1) \right) + c_j, \quad (4)$$

where N_0 is the noise spectral density; g_{ij} is the channel gain that is defined as $g_{ij} = k \cdot \alpha \cdot |h_i|^2$, where $k = -1.5/(\log(5\text{BER}))$, α is the path loss, and $|h_i|$ is the fading channel gain [14]. The parameters δ_j and c_j are device-specific parameters that depends on networks interfaces.

[1]Although the proposed work considers the encoding model of the EEG signals, without loss of generality, it can be easily extended to consider the models of diverse biosignals and vital signs.

III. Distributed Optimization Framework

The eventual goal of our network slicing framework is to create end-to-end network slices that meet all the required KPI by different services. This can be achieved through implementing the following phases at the the centralized-SDN controller and LSDN controller (see Figure 1):

Inter-slice Allocation: It aims to configure the network slices to meet all services' requirements, while minimizing the total cost of using network and computation resources. To obtain this goal, the centralized-SDN controller will have to: (i) extract service requirements (such as service delay and reliability), and (ii) reserve virtualize network resources and functions. Thus, the centralized-SDN controller will have to make joint decisions on the VNFs placement, resource reservation for diverse network slices, and traffic routing, such that the overall cost is minimized.

Intra-slice Allocation: The LSDN controller is distributing the resources of each slice to its attached EUs (RAN slicing), considering EUs' requirements and virtual resources availability.

In what follows, we present the formulated optimization problems at the centralized-SDN and LSDN controllers.

A. Inter-slice allocation problem

Reserving network resources with minimum cost for each slice is the main concern for service virtualization and network slicing. Hence, it is important to model such cost, considering: (i) the creation cost $c_n(v)$ of a VNF instance v at node n (this cost is null if the VNF instance is already exist and can be reused); (ii) the computational cost $c_n(k)$ of using a unit resource k at node n; (iii) the communication cost $c(i,j)$ of sending a unit traffic over link (i,j).

Upon receiving a request to deploy a service instance s, the centralized-SDN controller works on solving the inter-slice allocation problem, which is formulated as a cost-minimization problem, as fellows:

$$\textbf{P1:} \quad \min_{S_{f,p}, a_n(f,k)} (U_1) \tag{5}$$

such that

$$S_{f,p}\left[d_n(f,p) + d_p(f,p)\right] \leq D_f, \quad \forall f \in \mathcal{F}, \forall p \in \mathcal{P} \tag{6}$$

$$\prod_{i,j} \eta_j(t)\eta_{i,j}(t) \geq S_{f,p} \cdot H_f, \forall f \in \mathcal{F}, \forall p \in \mathcal{P} \tag{7}$$

$$\sum_{f \in \mathcal{F}} d_{f,p,l} \cdot r_{f,l} \leq W_l, \quad \forall l \in \mathcal{L}, \tag{8}$$

$$B(f,n,n+1) = B(f,n-1,n) \cdot \kappa(f,n), \quad \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{9}$$

$$\sum_{l \in p} d_{f,p,l} = N_p \cdot S_{f,p}, \quad \forall f \in \mathcal{F}, \forall p \in \mathcal{P}, \tag{10}$$

$$\sum_{p \in P} S_{f,p} = 1, \quad \forall f \in \mathcal{F}, \tag{11}$$

$$d_{f,p,l} \in \{0,1\}, \quad \forall f \in \mathcal{F}, \forall p \in \mathcal{P}, \forall l \in \mathcal{L}, \tag{12}$$

$$S_{f,p} \in \{0,1\}, \quad \forall f \in \mathcal{F}, \forall p \in \mathcal{P}, \tag{13}$$

where

$$U_1 = \sum_f \sum_p S_{f,p} \cdot \left[\sum_n \sum_v c_n(v) + \sum_n \sum_k c_n(k) a_n(f,k) + \sum_{(i,j)} c(i,j) B(f,i,j) \right].$$

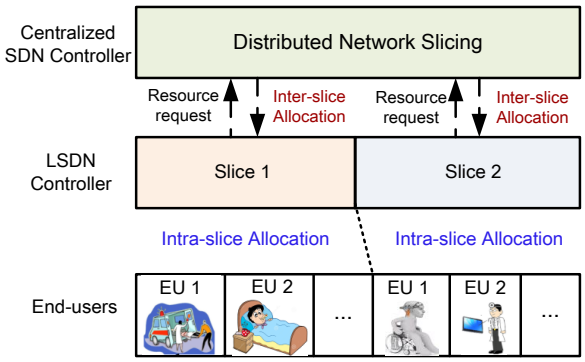

Fig. 1. Diagram representing the proposed network slicing framework, highlighting the different tasks performed by the centralized-SDN controller and LSDN controller.

The objective of **P1** is to reserve the required resources for each network slice (or service flow) and traffic route, such that the total cost is minimized while fulfilling all KPIs of diverse services. In (5), we define a path indicator $S_{f,p}$ that represents the selection decision of path p by flow f, where $S_{f,p} = 1$ when path p is selected, otherwise it will be zero. Thus, the unknowns in this problem are $S_{f,p}$ and $a_n(f,k)$, i.e., each flow (or service) needs to determine its transfer path and the amount of resources to be reserved in all network nodes along this path. Moreover, as mentioned in Section II, delay, reliability, and links' capacity constraints, i.e., (6)-(8), must be fulfilled for all flows, where $r_{f,l}$ is the data rate of flow f over link l. Constraint (9) represents the flow conservation constraint after processing a traffic flow f at a VNF v in node n. Constraint (10) instead ensures that once path p is selected by flow f all links along this path is reserved for this flow, where $d_{f,p,l}$ is the link selection indicator of link l over path p, and N_p is the number of hops in path p. Constraint (11) ensures that only one path will be selected by each flow.

We remark that the inter-slice allocation phase operates in a large timescale, which means that the problem in **P1** will be run after a long period. However, during this period, the centralized-SDN controller monitors the performance of diverse network slices, leveraging feedback information (e.g., resource utilization and service level agreement satisfaction) coming from the underlying network, in order to adjust slices' configuration in the next period.

B. Intra-slice allocation problem

After the creation and configuration of the network slices in the inter-slice allocation phase, the allocated resources for each slice should be distributed to its attached EUs, which is the main objective of the intra-slice allocation phase. Intra-slice allocation process operates in a smaller timescale (e.g., 100 ms [15]). The allocated resources include optimizing EU association with the most appropriate radio access points (e.g., micro BSs or femto BSs) and orchestrating network resources for EUs (i.e., throughput and computational resources). However, for optimizing the performance of different EUs, the LSDN controllers should inherently consider multiple, conflicting objectives of the EUs. Indeed, the proposed mathematical model becomes more realistic and efficient if different evaluation aspects, such as energy consumption, delay, and applications' QoS requirements, are explicitly considered as objective functions. Thus, with the aid of multi-objective optimization problem, we define

a single aggregate objective function to convert the multiple objectives into a single objective function, as follows:

$$U_2 = \sum_{i=1}^{i=N_u} \sum_{j=1}^{M} s_{ij} \cdot \left[\alpha_i \cdot \tilde{E}_{ij} + \beta_i \tilde{d}_{ij} + \gamma_i \cdot \psi_i \right], \quad (14)$$

where $\tilde{E}_{ij}$, $\tilde{d}_{ij}$, and ψ_i are the normalized energy, delay, and distortion of EU i over RAN j, respectively. However, these objectives represent different ranges and units of measurement, hence we first normalize them with respect to their maximum values to make them dimensionless and comparable. Also, it is assumed that EUs' devices are battery-operated devices, hence it is important to consider the transmission energy consumption while ignoring the processing energy that mainly consumed at the edge/RAN level, which have fixed power sources. In (14), s_{ij} is the access point selection indicator, where $s_{ij} = 1$ when an EU i selects a radio access point j(or RAN j for the sake of brevity), otherwise it will be zero. Also, it is assumed that each EU i ($i = 1, \cdots N_u$) can be associated with one of the available M access point/RANs. The weighting coefficients α_i, β_i, and γ_i represent the relative importance of the three objectives for each EU, such that $\alpha_i + \beta_i + \gamma_i = 1$. These weighting coefficients can play an important role to map different EU' conditions/preferences. Consequently, the intra-slice allocation problem is formulated as:

$$\textbf{P2:} \quad \min_{s_{ij}, \kappa_i, a_{ij}, r_{ij}} (U_2) \quad (15)$$

such that

$$\sum_{i=1}^{i=N_u} s_{ij} \cdot r_{ij} \leq R_j, \quad \forall j \in M, \quad (16)$$

$$\sum_{i=1}^{i=N_u} s_{ij} \cdot a_{ij} \leq a_j, \quad \forall j \in M, \quad (17)$$

$$C_{i,j} \leq C_{th}(i), \quad \forall i \in N_u, \quad (18)$$

$$\sum_{j=1}^{i=M} s_{ij} = 1, \quad \forall i \in N_u, \quad (19)$$

$$s_{ij} \in \{0, 1\}, \; r_{ij} \geq 0, \quad \forall i \in N, \forall j \in M, \quad (20)$$

$$0 \leq \kappa_i \leq 1, \quad \forall i \in N. \quad (21)$$

The unknowns in this problem are s_{ij}, κ_i, a_{ij}, and r_{ij}, i.e., each EU needs to determine its selected RAN, compression ratio, and the amount of resources to be reserved in this RAN, in terms of the computational resources and throughput. We remark that the variables $a_{ij}(v, k)$ refer to the quantity of resources of type k that are assigned to the instance of VNF v deployed at RAN j and used for the traffic generated at EU i, however for ease of presentation, we refer to them by a_{ij}. The available budget or cost constraint at each EU is represented by (18), where $C_{i,j}$ is the cost of using network and computation resources, and $C_{th}(i)$ is the maximum cost that can be paid by EU i. Constraint (16) and (17) instead represent the network capacity constraint and computing resources constraint, respectively, where R_j and a_j are the maximum network capacity and resources, respectively, while r_{ij} and a_{ij} being the available throughput and computing resources that can be used by EU i over RAN j (i.e., communication and computational resource share), respectively. Finally, constraint (19) ensures that at least one RAN has been selected by each EU.

IV. Proposed Solution

In this section, we present our solution for the inter-slice allocation problem using Inter-Slice Allocation (ISA) algorithm, while solving the intra-slice allocation problem using distributed optimization.

A. Inter-slice Allocation

Typically, the problem of jointly making resource allocation (or VNF placement) and traffic routing is hard; even simpler version of this problem (with only one KPI) has been proven to be NP-hard problem [16]. Also, the formulated inter-slice allocation problem in **P1** can be seen as a more complex version of a multi-constrained path problem, where the costs associated with different links are changing at every hop [17]. Hence, directly solving **P1** is unpractical, which motivates us to propose an efficient solution, called ISA, for which we could: (i) obtain effective resource allocation and traffic routing decisions, whose feasibility is guaranteed; (ii) the obtained decisions are close to the optimal and can be made in polynomial time.

In the proposed ISA algorithm, we opt to reduce the complexity of **P1** by identifying first the optimum traffic route for each flow, then optimizing the allocated resources over this route, such that the provided constraints are satisfied. Thus, the problem **P1** is simplified into the following resource allocation problem:

$$\textbf{P1-S:} \quad \min_{a_n(f,k)} \left(\tilde{U}_1 \right) \quad (22)$$

subject to (6), (9),

where $\tilde{U}_1$ is the objective function in (5) for a specific flow f and path p. The main steps of the proposed ISA algorithm include: (i) determine all possible paths that fulfill reliability and network capacity constraints for each service flow; (ii) for each path, solve the optimization problem in (22) to obtain the allocated resources and costs for it; (iii) select the path with minimum cost, for each service flow. This algorithm is considered optimal since it searches all possible paths for each service flow, and selects the minimum cost path that fulfills the services' requirements. The details of ISA algorithm are illustrated in Algorithm 1.

Algorithm 1 Inter-Slice Allocation (ISA) Algorithm

1: **Input:** $c_n(v)$, $c_n(k)$, $c(i,j)$, $r_{f,l}$, W_l, D_f, and H_f
2: **for** $f = 1$ to $|\mathcal{F}|$ **do**
3: Determine the list of all possible paths that fulfill the constraints in (7), (8).
4: For each path p in the list, solve the problem in (22).
5: Obtain optimal path p^* that minimize the aggregate cost $\tilde{U}_1$.
6: Set $S_{f,p^*} = 1$, and allocate the resources $a_n(f,k)$ for the selected path p^*.
7: **end for**
8: **Output:** S_{f,p^*}, $a_n(f,k)$

B. Intra-slice Allocation

The formulated intra-slice allocation problem in **P2** is a mixed-integer programming problem (i.e., NP-complete), due to the non-linearity of the objective function and the existence of the constraints in (19) and (20). Hence, solving the problem using convex optimization tools, or transforming it into a Geometric Program would not work

in this case. Thus, this problem will be analytically solved in two steps: (i) EU optimization, to find the selected RAN and compression ratio for each EU, and (ii) network optimization, for resource allocation at the network level. The first step will be a function of EUs' variables (or local variables), i.e., s_{ij}'s and κ_i's, while the second step being a function of network level variables (or global variables), i.e., a_{ij}'s and r_{ij}'s.

1) EU optimization: In this step, each EU will obtain its selected RAN and compression ratio, given that each RAN has allocated fixed amount of resources to different EUs, by solving the following EU optimization problem:

$$\textbf{P2-EU:} \quad \min_{s_{ij},\kappa_i} \sum_{j=1}^{M} s_{ij} \left[\alpha_i \tilde{E}_{ij} + \beta_i \tilde{d}_{ij} + \gamma_i \psi_i\right] \quad (23)$$
$$\text{subject to (18)-(21).}$$

Again, the problem in (23) is still neither a Linear Programming (LP) problem nor a convex problem [18]. However, it can be solved by decomposing it into two sub-optimization problems, for which an optimal, analytical solution is obtained. The decision variables in (23) are RAN selection variables s_{ij}'s, which affect the energy consumption and delay terms in the objective function, and adaptive compression variables κ_i, which affect the obtained distortion and transmitted data length. Hence, each EU can optimize its compression variable independently from the selected RAN selection, and vice versa. Thus, to solve the problem in (23), we proceed as follows.

For given κ_i, the formulated problem in (23) is turned to be LP; hence the optimal s_{ij} is calculated by maximizing the values of s_{ij}'s for which the corresponding objective function is minimum. Then, to fulfill the constraint in (18), all RANs with a higher cost than $C_{th}(i)$ is eliminated. Also, in order to fulfill the constraints in (19) and (20), the optimal s_{ij} is calculated as:

$$s^*_{ij} = \begin{cases} 1, & j = \arg\min_{s_{ij}} \left(\alpha_i \tilde{E}_{ij} + \beta_i \tilde{d}_{ij} + \gamma_i \psi_i\right) \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

As far as s^*_{ij} is obtained, a closed-form expression for κ_i can be derived by imposing that the derivative with respect to κ_i of the objective function in (23) is equal to 0. Hence, the optimal κ_i is calculated as:

$$\kappa^*_i = 1 - \left(\frac{100 \sum_{j=1}^{M} s_{ij} \cdot \bar{U}_{ij}}{\gamma_i x_2 x_3}\right)^{-\frac{1}{1+x_3}}, \quad (25)$$

where $\bar{U}_{ij}$ is the summation of weighted energy consumption and delay without compression, i.e., $\bar{U}_{ij} = \alpha_i \tilde{E}_{ij} + \beta_i \tilde{d}_{ij}$ at $\kappa_i = 0$.

2) Network optimization: After calculating the EUs' variables in (24) and (25), each EU send a *Request To Send* (RTS) message to the selected RAN with its local parameters. i.e., transmitted data length B_i, and available budged $C_{th}(i)$. Based on these requests, each RAN allocates the communication and computational resources r_{ij}, a_{ij}, respectively, for the associated EU $\tilde{N}_u$ by solving the following network optimization:

$$\textbf{P2-RAN:} \quad \min_{r_{ij},a_{ij}} \sum_{i=1}^{\tilde{N}_u} s_{ij} \left[\alpha_i \tilde{E}_{ij} + \beta_i \tilde{d}_{ij} + \gamma_i \psi_i\right] \quad (26)$$
$$\text{subject to (16)-(18).}$$

Now, the problem in (26) is turned to be convex, thus it can be solved efficiently using different convex optimization tools [18]. If the allocated resources to an EU i is sufficient, the connection request is admitted and the RAN sends *Clear To send* (CTS) message to the EU. Otherwise, the request is rejected. In case of rejection, the EU will have to re-run its optimization to select another RAN.

V. Simulation Results

In this section, we consider the network topology and simulation parameters shown in Figure 2, where two service flows for a practical scenario of remote monitoring applications, i.e., wireless brain monitoring, are considered.

Figure 3 assesses the performance of the proposed ISA algorithm compared to a baseline algorithm, called Hop-based Solution (HBS), while increasing the maximum delay deadline of flow 1 and flow 2. The latter refers to a solution that reduces the complexity of solving **P1** by selecting, at each node, the next hop with minimum cost. As shown, ISA algorithm outperforms HBS by always selecting the optimal path with minimum cost. This enhancement can be significant in case of large networks with multi-hops. The intuitive explanation is that the proposed ISA minimizes the end-to-end cost of diverse flows while considering all possible paths from sources to the destinations; unlike HBS that takes local decisions at each intermediate node to select the next hop with minimum cost. In this figure, it is assumed that the transmitted data length of each flow is 25 Kb.

In Figure 4 and Figure 5, we compare the performance of the proposed LSDN-DS against User-based Distributed Solution (UDS), which implements the idea of RAN selection in [19]. Herein, four RANS are considered to be available for each flow to transmit its data with arbitrary throughput $R_1 = 2.15$ Mbps, $R_2 = 1.6$ Mbps, $R_3 = 1$ Mbps, and $R_4 = 0.9$ Mbps, respectively, while assuming the transmitted data length of flow 1 and 2 is 50 Kb and 70 Kb, respectively. Figure 5 presents the objective function variations in **P2** as a function of the throughput increase of different RANs, while Figure 4 depicting the value of different performance metrics, i.e., energy consumption,

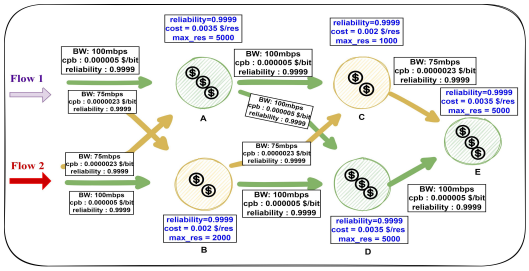

Fig. 2. The considered network topology and simulation parameters.

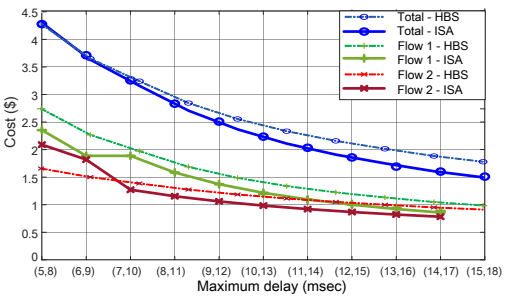

Fig. 3. Cost variations of the inter-slice allocation problem for ISA and HBS, while increasing the maximum delay deadline of flow 1 and flow 2.

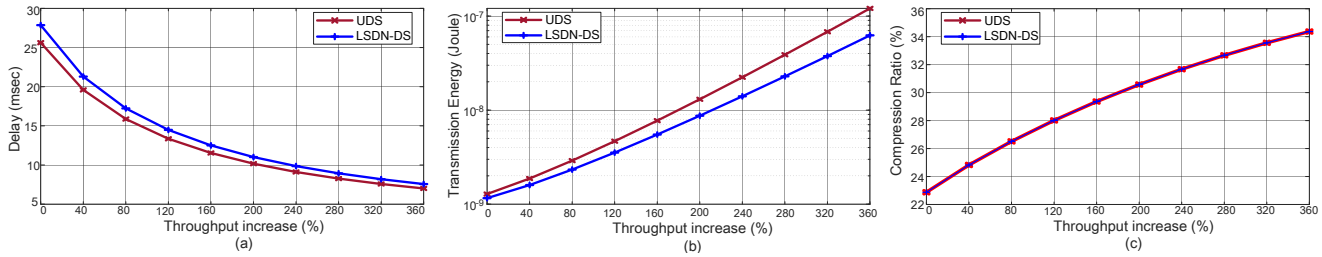

Fig. 4. A comparison of different performance metrics under LSDN-DS and UDS, while increasing the throughput.

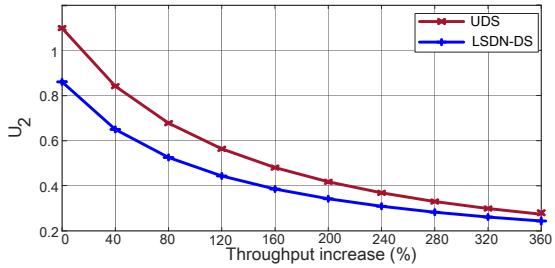

Fig. 5. Objective function variations of the intra-slice allocation problem under LSDN-DS and UDS, while increasing the throughput.

delay, and compression ratio. UDS assumes that each EU solves its own optimization problem, to find the selected RAN and compression ratio, while allocating the network resources, by different RANs, for the associated EUs using proportional fair algorithm. On the contrary, our LSDN-DS optimizes both the EU and network behaviors by solving the EU optimization in (23) and network optimization in (26), respectively. As shown, increasing the throughput at different RANs (i.e., increasing the available resources) decreases the objective function U_2 (see Figure 5), which is mainly due to increasing the EUs' data rate, hence decreasing the delay (see Figure 4-(a)). However, increasing EUs' data rate comes also at the expense of increasing the energy consumption, which leads to increasing the compression ratio to decrease the transmitted data length (see Figure 4-(b),(c)). Hence, optimizing the intra-slice allocation using our LSDN-DS could outperform the performance of UDS that considers only EU optimization.

VI. Conclusion

This paper presents a dynamic network slicing framework that aims to optimize the performance of diverse supported services, while considering services' characteristics and KPIs, e.g., availability, reliability, and data quality. In particular, we present a novel distributed optimization methodology for inter-slice allocation and intra-slice allocation, in order to optimize the selection of radio points of access, resource allocation, and data routing. The proposed solutions could obtain the best trade-off between diverse performance metrics, while outperforming the baseline approaches that consider partial network view or fair resource allocation.

Acknowledgement

This work was made possible by NPRP grant # NPRP13S-0205-200265 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

References

[1] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Commun. Stand. Mag.*, vol. 2, no. 1, 2018.

[2] A. Awad, A. Mohamed, C.-F. Chiasserini, and T. Elfouly, "Network association with dynamic pricing over d2d-enabled heterogeneous networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[3] F. Z. Yousaf, M. Gramaglia, V. Friderikos, B. Gajic, D. Von Hugo, B. Sayadi, V. Sciancalepore, and M. R. Crippa, "Network slicing with flexible mobility and qos/qoe support for 5g networks," in *IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2017, pp. 1195–1201.

[4] G. Einziger, M. Goldstein, and Y. Sa'ar, "Faster placement of virtual machines through adaptive caching," in *IEEE INFOCOM*, 2019.

[5] M. Condoluci and T. Mahmoodi, "Softwarization and virtualization in 5g mobile networks: Benefits, trends and challenges," *Computer Networks*, vol. 146, pp. 65–84, 2018.

[6] B. Ojaghi, F. Adelantado, A. Antonopoulos, and C. Verikoukis, "Slicedran: Service-aware network slicing framework for 5G radio access networks," *IEEE Systems Journal*, pp. 1–12, 2021.

[7] M. Bouet and V. Conan, "Mobile edge computing resources optimization: A geo-clustering approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 787–796, 2018.

[8] V. Sciancalepore, F. Z. Yousaf, and X. Costa-Perez, "z-TORCH: An automated NFV orchestration and monitoring solution," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 4, pp. 1292–1306, 2018.

[9] J. Martín-Pérez et al., "Okpi: All-kpi network slicing through efficient resource allocation," in *IEEE INFOCOM*, 2020, pp. 804–813.

[10] K. Kamran, E. Yeh, and Q. Ma, "Deco: Joint computation, caching and forwarding in data-centric computing networks," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 111–120.

[11] A. A. Abdellatif, A. Mohamed, and C.-F. Chiasserini, "User-centric networks selection with adaptive data compression for smart health," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3618–3628, 2018.

[12] A. A. Abdellatif et al., "Edge computing for energy-efficient smart health systems: Data and application-specific approaches," in *Energy Efficiency of Medical Devices and Healthcare Applications*. Elsevier, 2020, pp. 53–67.

[13] A. Awad, M. Hamdy, A. Mohamed, and H. Alnuweiri, "Real-time implementation and evaluation of an adaptive energy-aware data compression for wireless EEG monitoring systems," *QSHINE*, pp. 108–114, Aug. 2014.

[14] A. Awad, O. A. Nasr, and M. M. Khairy, "Energy-aware routing for delay-sensitive applications over wireless multihop mesh networks," in *International Wireless Communications and Mobile Computing Conference*. IEEE, 2011, pp. 1075–1080.

[15] W. Wu, C. Zhou, M. Li, H. Wu, H. Zhou, N. Zhang, W. Zhuang, et al., "Ai-native network slicing for 6G networks," *arXiv preprint arXiv:2105.08576*, 2021.

[16] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, "Approximation algorithms for the NFV service distribution problem," in *IEEE INFOCOM*, 2017, pp. 1–9.

[17] G. Xue, A. Sen, W. Zhang, J. Tang, and K. Thulasiraman, "Finding a path subject to many additive qos constraints," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 201–211, 2007.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. cambridge university press, 2003.

[19] Z. Chkirbene, A. Awad, A. Mohamed, A. Erbad, and M. Guizani, "Deep reinforcement learning for network selection over heterogeneous health systems," *IEEE Trans. Netw. Sci*, pp. 1–1, 2021.