IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 12, 2020, accepted November 29, 2020, date of publication December 15, 2020,
date of current version December 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3044951

Novel NFV Aware Network Service for Intelligent Network Slicing Based on Squatting-Kicking Model

AHMED EL-MEKKAWI, XAVIER HESSELBACH, (Senior Member, IEEE), AND JOSE RAMON PINEY

Department of Network Engineering, Universitat Politecnica de Catalunya, 08034 Barcelona, Spain

Corresponding author: Ahmed El-Mekkawi (ahmed.mohamed.abdelaty.elmekaw@upc.edu)

This work has been supported by the Ministerio de Ciencia e Innovacion of the Spanish Government under Project PID2019-108713RB-C51.

ABSTRACT Future networks starting from 5G will depend on network slicing to meet a wide range of network services (NSs) with various quality of service (QoS) requirements. With the powerful Network Function Virtualization (NFV) technology available, network slices can be rapidly deployed and centrally managed, giving rise to simplified management, high resource utilization, and cost-efficiency. This is achieved by realizing NSs on general-purpose hardware, hence, replacing traditional middleboxes. However, realizing fast deployment of end-to-end network slices still requires intelligent resource allocation algorithms to efficiently use the network resources and ensure QoS among different slice categories during congestion cases. This is especially important at the links of the network because of the scarcity of their resources. Consequently, this paper proposes a paradigm based on NFV architecture aimed at providing the massive computational capacity required in the NSs and supporting the resource allocation strategy proposed for multiple slice networks based on resources utilization optimization using a proposed and analyzed Squatting-Kicking model (SKM). SKM is a suitable algorithm for dynamically allocating network resources to different priority slices along paths and improving resource utilization under congested scenarios. Simulation results show that the proposed service deployment algorithm achieves 100% in terms of both overall resource utilization and admission for higher priority slices in some scenarios in bandwidth-constrained contexts, which cannot be achieved by other existing schemes due to priority constraints.

INDEX TERMS 5G, network slicing, network services, quality of service, virtual network function, SKM.

I. INTRODUCTION

In network slicing of future networks starting from 5G, the intent is to take infrastructure resources from the spectrum, antennas and all of the backend network and devices and use them to realize multiple sub-networks with different properties. Each sub-network slices the resources from the physical network End-to-End (E2E) to realize its own independent, no-compromise network for its favored applications. The Third-Generation Partnership Project (3GPP) has defined four main slices types [1], [2]: the enhanced Mobile Broadband (eMBB) targets to meet ultra-high data rates as required for 4K or immersive 3d video; the Massive Internet of Thing (MIoT) is targeted for devices that require massive connections like agriculture; the ultra-reliable low latency communications (URRLC) is targeted for ultra-low latency and high-reliability services like self-driving vehicles; and the vehicle-to-everything (V2X) is targeted for advanced driving assistance services and needs ultra-low latency and high data rates. Moreover, the architecture are adaptable to the different slice types that may emerge in the future. Because it would be far too costly to allocate a complete E2E network to each type of slice, the network infrastructure that promotes 5G will employ sharing techniques (virtualization technologies such as NFV), which allow for multiple slice types to coexist without having too many resources [3], [4]. With NFV,

The associate editor coordinating the review of this manuscript and approving it for publication was Zehua Guo.

network resources can be efficiently allocated and the process of implementing user-oriented services accelerated which saves both cost and time by enabling the implementation and deployment of middleboxes as virtual network functions (VNFs) running on Virtual Machines (VMs). In other words, by using NFV paradigm, network slices associated with resources can ensure optimization of resource provisioning to the end-users with high quality of service (QoS) and guarantee the performance of VNFs operations, including minimum latency and failure rate. Moreover, NFV simplifies service deployment by exploiting the concept of service chaining [5]–[10].

Virtualization and progressive softwarization of network function in NFV architecture give rise to new opportunities for improving application tools and platforms in the market, like management and orchestration (MANO), for controlling the life-cycles of the slices and as well as the underlying VNFs at the network levels; for instance, European Telecommunications Standards Institute (ETSI) standardizes the VNF structure [11] and suggests the OpenSource MANO (OSM) [12] platform. These platforms can undoubtedly facilitate the sharing of resources between slices, but they still require intelligent resource allocation algorithms to permit a particularised slice to meet its service level agreement (SLA) such as QoS and bandwidth. Moreover, such intelligent algorithms can improve load balancing, resource utilization, and network performance.

Realizing an E2E slice (e.g. Access, core, Transport, Backhauls) entails resource provisioning across both nodes and links of the network infrastructure as shown in Fig. 1. However, the management of link resources is a more critical aspect of the network slicing problem compared to node resource management. This is due to the complexity related to bandwidth allocation along a path and management of the prioritization on the links, among others, while jointly ensuring that the routing paths meet the QoS constraints under limited network resources [13]–[17]. In addition, it is difficult to select an optimal path for different priority demands and subsequent allocation of resources from source to destination in the physical substrate network.

Therefore, optimal resource allocation algorithms are vital to optimize link resource utilization, meeting various QoS requirements of services and providing high admission for higher priority slices under congested scenarios [18], [19]. Currently, few works, in literature focus on the link resources management (e.g., bandwidth management) considering a multi-slice scenario with prioritized demands despite its necessity for the realization of future network services characterized by massive bandwidth requirements [20]–[22]. Regarding bandwidth resource management under a multi-slice scenario, SKM exhibits competitiveness compared to Bandwidth Allocation Models (BAMs) and best-effort algorithms, especially during congested scenarios [23]. We, therefore, consider the SKM algorithm for this work.

In this work, we develop an algorithm that uses the intelligence of SKM strategy for efficient deployment and

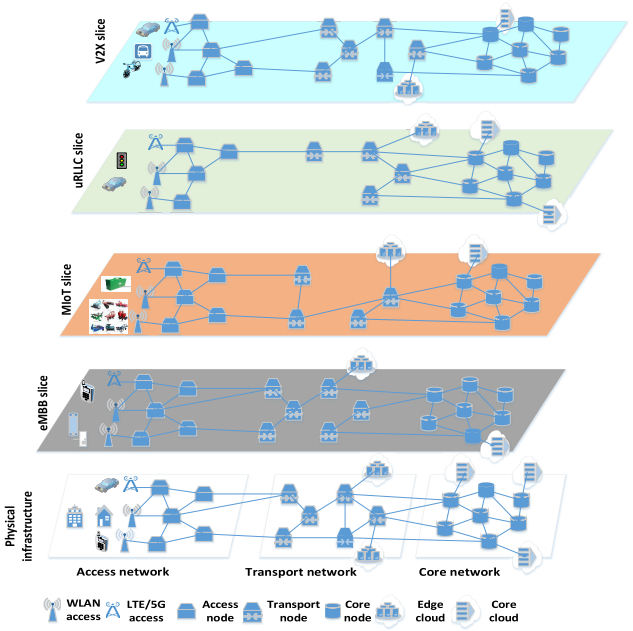


FIGURE 1. E2E 5G network slicing architecture.

allocation of network resources in a multi-slice scenario while aiming at maximizing the utilization by choosing less congested paths based on the computation algorithms executed in the NFV architecture. We formally define the proposed algorithm to solve the problem of real-time resource allocation for QoS E2E routing considering realistic network behavior. This is carried out by incorporating strict constraints such as priority and bandwidth, and considering full network topology under online and offline demand arrival. Cognizant of this fact, this paper focuses on how intelligence can be deployed in NFV in order to provide efficient utilization of link bandwidth resources in a multi-service scenario considering strict constraints as required in 5G networks. Moreover, we want to point out that other parameters such as delay can be integrated into our proposed algorithm. SKM has squatting and kicking techniques. The squatting technique allows higher priority slices of service to utilize resources reserved for lower priority ones when being unused and vice versa. For higher priority slices, it is intended to improve the utilization, increase the acceptance ratio of the demands, and guarantee no rejection of demands when there are any unutilized resources in the network. On the other hand, the kicking technique guarantees better QoS of higher priority traffic, where the higher priority slices can kick lower priority ones from their currently allocated resources. The results demonstrate that our proposed algorithm strictly prioritizes higher priority slices in congested scenarios while resulting in similar performance compared to other algorithms using BAMs in the non-congested scenarios.

The rest of the paper is organized as follows: Section II, lists the current limits in 5G networks and positions our contribution. Section III presents related work in literature.

In section IV the network model and problem formulation are presented. In section V, we present and describe our proposed algorithm for the intelligent management of the multi-slice network. Section VI describes the performance evaluation of our algorithm. Finally, section VII concludes the paper and proposes recommendations for future work.

II. ISSUE AND POSITIONING OF OUR CONTRIBUTION

In this paper, the authors are exploiting the results from their previous work in El-mekkawi *et al.* [23] in the following aspects:

- Proposing a service deployment algorithm based on the intelligence of the SKM model defined on [23] to maximize the number of successfully allocated service demands while maximizing the utilization and uniformly distributing the traffic across the different links of the network. Moreover, this algorithm can handle service requirements with different strict constraints, in both off-line and on-line modes.
- Adopting a realistic 5G network environment: the work proposed in [23] was analysed based on scenarios that are not representative of a realistic 5G network environment by considering single link network topology. However, in this work, we have analysed based on a realistic 5G scenario where the network topology is complex, the transmission is real-time, the requests arrive in online mode, and the demand structure is much more complicated compared to the mentioned paper (i.e., the demands are defined by source, destination, bandwidth, priority and lifetime to be allocated along the requested in a given network). Hence, finding routing paths in a given network and allocating/reserving the resources along the path should be considered, which makes it more complex than the setup of the mentioned paper. In addition, the online arrival of requests makes it imperative to keep the status of the substrate network resources always up-to-date, in order to directly assess the probabilities of allocating other requests as they arrive. This is more complicated in a full network topology setting.
- The performance of the proposed algorithm is analyzed by not only representative examples, but also long simulations that vary in terms of the system parameters as well as using topologies and metrics.

The proposed solution aims at smartly sharing resources among different slices of services according to their service needs and assigned priorities. 5G network management with network slicing scenario faces numerous challenges that can be addressed through our proposed model including but not limited to the following:

- Admission control and resource allocation problems:
  With the fast growth of 5G networks demands, limitation of resources and the QoS requirements of users, it is essential to know the maximum number of users that can be admitted simultaneously to the system while efficiently using the available resources and satisfying the QoS requirements [1]. Thus, we are motivated to find a solution for admission control and resource allocation problems in 5G networks that can achieve a targeted level of QoS by efficiently sharing the same resources between different priority application scenarios users.
- The need for a preemption policy:
  The 5G networks should allow a flexible means to enforce prioritization among the services under congested cases. The traffic prioritization may be enforced by adjusting resource utilization or preempting lower priority traffic [2]. In this regard, in this work, we exploit the kicking technique defined in [23] to ensure E2E high admission for higher priority slices during congested cases.
- E2E QoS for a service:
  Setting up QoS policies in nodes along the path is a complicated task. However, another constraint must be addressed; it is the routing with QoS constraints. In other words, how to ensure that the used path meets the user' bandwidth requirements [24]. SKM strives to find the most suitable path to meet the QoS requirements of users through differentiation of traffic slices and resource allocation using squatting and kicking techniques.
- Suitable strategy for future application scenarios:
  Our solution is deployable in any queue-based system. However, the benefits of our algorithm are more distinctive for future networks starting from 5G characterized by massive bandwidth requirements with prioritization under limited network resources in congested and extreme scenarios such as emergency and disasters. This is even more significant in a network slicing environment.

In summary, our contributions are the following:

1) An intelligent service deployment algorithm that uses SKM strategy to jointly maximize resource utilization, acceptance rate and ensure QoS for higher priority slices while meeting various service constraints in a multi-slice scenario. The algorithm is defined mathematically considering a real-time application for full network topology with strict constraints demand such as priority and bandwidth. The algorithm proposed takes into account QoS management and QoS constraint routing with autonomic features and feasible computation time. Moreover, the proposed algorithm can be adapted to different constraints, topologies and scenarios.
2) The proposed algorithm provides a novel policy for E2E network slicing deployment based on efficient selection and serving demands. This policy takes into account QoS constraints for different priorities/slices. It is also suggested for optimizing the behavior of network slicing using intelligent mechanism that is proposed to be adopted in NFV architecture. Moreover, it acts as an admission control function to ensure proper performance of QoS levels while increasing the overall use of resources across the entire substrate network.

3) Performance evaluations and analyses of the proposed algorithm are presented against service deployment algorithms incorporating BAM strategies in terms of several metrics. These metrics aim at reflecting the algorithm ability to manage multi-slice demands under various input traffic loads in a resource-limited 5G network. Moreover, we compared our proposed algorithm against the recent work from Reale et al [25].

III. RELATED WORK

In this section, we review recent works and briefly introduce studies on enabling technologies related to the mechanism of network slicing. We mainly focus on exiting works that relate to: (i) E2E network slicing; (ii) Define a new multi-path routing strategy that considers QoS constraints based on the NFV architecture and (iii) Resource allocation and QoS models.

A. E2E NETWORK SLICING

The concept of network slicing has been proposed to address the diversified service requirements in future networks (e.g., 5G networks). Network slicing is still a nascent paradigm and needs management and orchestration of network functions, mapping and service descriptions. In the existing literature, few works are conducted on the deployment of E2E network slicing despite its necessity for the realization of network slices as it provides the operators with the ability to customize networks and meet various service demands. Moreover, E2E slices need to be instantiated rapidly. In [10], the authors demonstrate network slicing as a service (NSaaS) and present a business model of NSaaS. The authors in [26] present PERMIT slice orchestration system, which is the first to deal with E2E slicing. In [27], the authors design various algorithms and measurement indicators to meet various service requirements for only two types of slices (rate constraints and delay constraints) and to solve the E2E maximum access rate of the entire system. However, these works still suffer from inefficiency in terms of deployment and management of E2E slices and need crucial and promising models to address the above challenges.

B. ADAPTIVE QoS ASSIGNMENT FOR MULTIPATH NETWORKS

Ensuring a right QoS level for multi-path networks is one of the significant challenges for multi-slice networks, especially where bandwidth resource consumption restrictions appear [28], [29]. The impact of multi-path and QoS can only actually be felt when real-time demands are routed over the network, especially in future large-scale networks, whose bandwidth is one of the main concerns. Notwithstanding the many multi-path routing methods, these remain limited when paths have asymmetric performances and demands are delicate to SLA violations. In [30], the authors propose a model of the adaptive and dynamic VNF allocation problem considering VNF migration. Moreover, they also consider service function chains (SFCs) with QoS constraints. On the other hand, the authors in [31] propose and evaluate the performance of an algorithm to allocate and compute optical bandwidth resources in an NFV environment to minimize their costs while taking into account the different costs of the Infrastructure Providers. In [32] the authors propose a methodical way to elastically tune the proper link and server usage of each demand, compute a proper routing path length, and decide whether to reuse resources for each function incrementally or not.

From the aforementioned works in terms of path selection, as it computes an optimal routing path while taking into account how the resources are accessed according to a predefined set of slices on multi-sliced networks. Our algorithm strictly consider priorities and significantly differentiates between them under congested scenarios to improve the utilization and provide high protection for higher priority slice users depending on traffic, thus guaranteeing QoS and SLA.

C. RESOURCE ALLOCATION AND QoS MODELS

In the existing literature, several works deal with the dynamic resource allocation for ensuring a given QoS level per class, controlling how the resources are accessed and optimizing utilization. In terms of managing bandwidth allocation, BAMs are of high value in the context of QoS assurance. These BAMs are based on the Maximum Allocation Model (MAM) [33], the Russian Doll Model (RDM) [34] and the AllocTC-Sharing Model (AllocTC) [35]. In [36], the authors proposed an algorithm based on RDM to support dynamic bandwidth allocation for Differentiated services (DS) classes and improve bandwidth efficiency by providing the triple-play services to share the bandwidth. The allocation of bandwidth is based on the classification and prioritization of service. The proposed algorithm is applied to Ethernet Passive Optical Network (EPON) and assigns the fairness factor and services priority to the required bandwidth of the request.

The common problem of the algorithms based on RDM is that the resources reservation is made from the bottom to top; the low priority traffic shares its resources with the higher priority traffic and not the reverse, [37]–[39]. The common problem of the algorithms based on MAM is that there is no ability for sharing resources among traffic classes [40]. i To improve resource utilization, the reservation of resources should be supported in both directions. To this end, the authors in [35], proposed a model called AllocTC, which allows sharing of the unused bandwidth of high bandwidth applications with low priority and vice versa. In [41], the authors assessed the effectiveness of the AllocTC model compared to the RDM model. The authors have also shown by simulation that AllocTC is more efficient in terms of maximizing the link use and that it is better adapted to elastic traffic and high bandwidth usage. In [42], the authors launch a new approach to switch autonomously between models (MAM, RDM, G-RDM, and AllocTC) based on a controller. Switching from one algorithm to another can be done in different metrics, for example, link use, congestion probability,

and packet number. In [25], the authors proposed the Generalized Bandwidth Allocation algorithm (G-BAM) that adopted the BAM approach to improve the efficiency and performance of the Multiprotocol Label Switching DS-Traffic Engineering (MPLS DS-TE) network in an autonomous way. In Bahnasse [24], the authors proposed a new SDN-based architecture following a new smart and dynamic model (smart Alloc) for allocating and controlling the QoS and routing with QoS restrictions for a DS-TE network. This model is based on RDM, and AllocTC approaches. Firstly, it aims at classifying flows based on their threshold severity (high, medium, and low). Whatever the priority of the flow belonging to the high threshold, the latter can benefit from the loans of the other categories. Secondly, to collect bandwidth from other categories and to determine the fairness index to allocate resources precisely to all flows while taking into account their priorities. Smart Alloc was implemented on a controller to manage QoS and routing for only the MPLS DS-TE networks. However, these models cannot ensure high admission for higher priority classes and give high network utilization at the same time.

In our previous works, El-mekkawi *et al.* [43], [44], we proposed the concept of SKM aiming at realizing a 100% overall network utilization while guaranteeing high admission for higher priority classes. In El-mekkawi *et al.* [43], an offline resource allocation strategy is proposed for EON embedding to improve the computational capacity. The proposed model considers priority classes and utilizes NFV architecture. The proposed algorithm is described, analyzed, and compared with existing models in terms of flexibility in resource allocation per class and prioritization of channel usage. Besides the fact that the performance of this work was not compared with the recent state-of-the-art work such as AllocTC under various scenario conditions, this work considers only offline scenario. Moreover, no formal description of the SKM strategy was given in this work. In Elmekkawi *et al.* [44], we introduced a new flexible admission control mechanism on a pool of resources based on squatting and kicking techniques (SKM) which can be employed under network slicing scenario. Moreover, the algorithm used in El-mekkawi *et al.* [44] only checks the capacity of the system (i.e., a pool of resources) to potentially accept the demand without considering the links. Therefore, it only focuses on satisfying the simple demands, where the demands are just admitted according to a pool of resources.

IV. NETWORK MODEL AND PROBLEM FORMULATION

This section is divided into three subsections: infrastructure network model, slice request model and problem formulation.

A. INFRASTRUCTURE NETWORK MODEL

The substrate network is modelled as a directed graph $G(X, L)$ where X and L denote the set of all substrate nodes and substrate links respectively. If such a connection exists, we use $l_{i,j} \in L$ to denote a single edge substrate link between substrate node i $\in X$ and substrate node j $\in X$. Each substrate link $l_{i,j}$ is characterized by i) Maximum link resources capacity R($l_{i,j}$); ii) Available link resources at a given time denoted by $R_a^t(l_{i,j})$; iii) Consumed link resources $R_z^t(l_{i,j})$ at time t; iv) A set of traffic slices assigned along the link are denoted by $CTs(l_{i,j})$, where $CT_N(l_{i,j})$ is the highest priority slice and $CT_1(l_{i,j})$ is the lowest priority slice; v) Actually allocated resources to slice c $S_c(l_{i,j})$, where $c \in [1, N]$; vi) Slice resource constraints $RC_c(l_{i,j})$. If such a path exists, we use $P_{s,r}^k$ to denote kth shortest path between source node s $\in X$ and destination node r $\in X$, where $k \in [1, K]$. $P_{set}^{s,r}$ denotes the set of all K shortest paths from node s to node r. V($P_{s,r}^k$) represents the set of feasible physical substrate nodes to map VNFs for $P_{s,r}^k$.

B. SLICE REQUEST MODEL

In our model, each request belonging to any kind of slice to be allocated in the substrate network is denoted by i) A source node s $\in X$; ii) A destination node r $\in X$; iii) The amount of resources required belonging to slice c, $d_w(CT_c)$, where demand w $\in$ [1, D]; iv) priority $c_{d_w} \in [1, 3]$ and v) lifetime interval t_{d_w}. Further, in this work, we assume that the request volume is the required number of link resources, thus the potential paths from source to destination are determined when the request arrives. A description of all parameters, decision variables and main metrics used in this article is provided in Table 1.

C. PROBLEM FORMULATION

We propose the resource allocation problem to maximize the network resource utilization by allocating all service demands of slices identified below in the appropriate substrate network resources. The problem is formulated subject to the link and slice constraints, considering service demands with different priorities and link capacity requirements. The slice requirements and the available link resources are the inputs to the resource allocation phase along the substrate network. The output is the best routing path for a given slice request that optimizes network resources usage while guaranteeing high acceptance rates for higher priority slices.

In this work, we considered three main application scenarios as defined by 3GPP. These are described below:

1) eMBB: This application scenario does not need a special QoS guarantee. Hence, this slice is adopted in this work to satisfy the service requests of the lowest priority.
2) MIoT: This application scenario is characterised by a massive number of connected devices, usually transmitting a relatively low volume of non-delay sensitive data. Hence, this application scenario is adopted in this work to satisfy the service requests of the intermediate priority.
3) uRLLC: This application scenario is more stringent on delay requirements. Hence, this application scenario is adopted in this work to satisfy the service requests of the highest priority.

Please note that in this study, to simplify the evaluation of our proposed algorithm, we assume that a network service

TABLE 1. Notation and variables.

Notation	Description
G	Directed graph of the physical network.
X	Substrate network nodes.
L	Substrate network links.
$RC_c(l_{i,j})$	Resource constraints for slice c also equal to maximum reservable resources for slice c in $l_{i,j}$.
$CT_c(l_{i,j})$	Priority slice c in $l_{i,j}$.
R($l_{i,j}$)	Maximum reservable resources for all slices together and is equal to link capacity.
$d_w(CT_c)$	The amount of resources (size) of demand w belonging to slice c.
t	time variable.
$R_a^t(l_{i,j})$	Available resource capacity on $l_{i,j}$ at time t.
$R_z^t(l_{i,j})$	The consumed resources capacity on $l_{i,j}$ at time t.
$X_w^{t,l_{i,j}}$	is the binary variable equal to 1 if the demand $w \in W$ is assigned resources at link $l_{i,j} \in L$, zero otherwise.
$P_{s,r}^k$	The kth shortest path from s to r for the demand.
$P_{set}^{s,r}$	Set of all Kshortest paths from source s to destination r in the network.
V($P_{s,r}^k$)	set of feasible physical substrate nodes to map VNFs for $P_{s,r}^k$.
t_{d_w}	The duration of demand w.
T	Duration of the simulation window in time units.
D_c	Total number of demands by slice c.
D	Total number of demands by all slices.
$S_c(l_{i,j})$	The actually allocated resources to slice c on $l_{i,j}$.
c_{d_w}	Priority of demand w.
BD	Number of blocked demands by all slices.
BD_c	Number of blocked demands by slice c.
AD	Number of accepted demands by all slices.
AD_c	Number of accepted demands by slice c.
μ	The mean value of the utilization of all links across the network.
P_{re}	Number of preempted demands in the whole network.
$SH_q(l_{i,j})$	Squatted resources from higher priority $slice_q$ on $l_{i,j}$.
$SL_q(l_{i,j})$	Squatted resources from lower priority $slice_q$ on $l_{i,j}$.
$K_q(l_{i,j})$	Kicked resources from lower priority $slice_q$ on $l_{i,j}$.
$Z(d_w)$	1 if demand w is successfully mapped.

demand is acceptable when the link resources are available along the requested path from the source node to the destination node.

The mathematical definition of the proposed algorithm is expressed as follows:

The ultimate goal is to maximize network resource utilization while meeting demand constraints. Therefore, the major goal can be expressed by

$$Max\ U(T) \tag{1}$$

U(T) is the utilization of the links along the network at each time window T. The link resource utilization is related to the ratio of link resources used to the link capacity averaged across all substrate links described as follows:

$$U(T) = \frac{1}{T}\sum_{t\in T}\frac{1}{L}\sum_{\forall l_{i,j}\in L}\sum_{w\in W}\frac{X_w^{t,l_{i,j}} * d_w(CT_c)}{R(l_{i,j})} \tag{2}$$

where $X_w^{t,l_{i,j}} \in [1,0]$ is a binary variable equal to 1 if resources are allocated to the request $w \in W$ on the link $l_{i,j} \in L$, zero otherwise. $d_w(CT_c)$ indicates the bandwidth resources required by the w request. T indicates the duration of the simulation window in time units. The total used resources at link $l_{i,j} \in L$ at any unit time t descried by:

$$R_z^t(l_{i,j}) = \sum_{w\in W} X_w^{t,l_{i,j}} * d_w(CT_c) \tag{3}$$

This goal is subject to the following:

1) link constraints:

$$\sum_{\forall l_{i,j}\in P_{s,r}^n} RC_c(l_{i,j}) \le R(l_{i,j}), \quad \forall t \tag{4}$$

$$Z(d_w) = R_a^t(l_{i,j}) \ge P_{s,r}^k * d_w(CT_c), \quad \forall t, \quad \forall l_{i,j} \in P_{s,r}^k, w \in [1, D] \tag{5}$$

Eq. (4) ensures that the maximum reservable bandwidth for a link $l_{i,j}$ at any time is less than or equal to the link capacity for that link. Eq. (5) specifies if request w is allocated at a given time. In other words, the demand will be successfully allocated from source to destination if all links along the path have more available resources than required.

2) Slice constraints:

To allocate the demand into a set of traffic slices for each link along the requested path, we use SKM model proposed in [23]. SKM's model contains two techniques; squatting and kicking techniques. Squatting technique helps in sharing of unused resources between higher and lower priority service slices while kicking technique ensures proper QoS for higher traffic priority slices by expelling lower priority slices from resources directly assigned to them. SKM performs four steps to allocate each demand, which are as follows:

Step 1 (MAM): Upon arrival of a demand $d_w(CT_c)$ belonging to slice c, the following constraints are checked:

$$S_c(l_{i,j}) \le RC_c(l_{i,j}) \tag{6}$$

$$\sum_{c=1}^{N} RC_c(l_{i,j}) = R(l_{i,j}) \tag{7}$$

Eq. (6) ensures that the resources needed to serve the already existing demands plus the new demand do not exceed slice resources constraint while Eq. (7), ensures that the total amount of slices resources constraints should be equal to $R(l_{i,j})$. If constraints are satisfied, $d_w(CT_c)$ is accepted. Otherwise, try step 2.

Step 2 (Squatting-High or RDM): Try to squat unused resources starting from the higher adjacent priority slice upwards until there are enough resources

to satisfy $d_w(CT_c)$. If the resources are enough, then accept $d_w(CT_c)$, otherwise, try step 3. Note that the total allocatable resources in $CT_c(l_{i,j})$ cannot exceed the slice resource constraint $RC_c(l_{i,j})$ plus all squatted resources from higher priority slices as in Eq. (8). Eq. (9) indicates that $SH_q(l_{i,j})$ is less or equal to the difference between the slice resource constraint and the minimum between the allocated and the reserved resources for the same slice. Note that the highest priority slice cannot use Squatting-High strategy.

$$S_c(l_{i,j}) \leq RC_c(l_{i,j}) + \sum_{q=c+1}^{N} SH_q(l_{i,j}) \quad (8)$$

$$SH_q(l_{i,j}) \leq RC_q(l_{i,j}) - min(S_q(l_{i,j}), RC_q(l_{i,j}) \quad (9)$$

Step 3 (Squatting-Low): Try to squat unused resources starting from the lower adjacent priority slice downwards until there are enough resources to satisfy $d_w(CT_c)$. If the squatted higher resources plus the squatted lower resources satisfy $d_w(CT_c)$, then accept $d_w(CT_c)$, otherwise, try step 4. Eq. (10) indicates that the total allocatable resources in $CT_c(l_{i,j})$ cannot exceed the slice resource constraint plus all squatted resources in both squatting high and low. Moreover, $SL_q(l_{i,j})$ works like $SH_q(l_{i,j})$, but from lower slices, as shown in Eq. (11). Note that the lowest priority slice cannot use Squatting-Low strategy.

$$S_c(l_{i,j}) \leq RC_c(l_{i,j}) + \sum_{q=c+1}^{N} SH_q(l_{i,j}) + \sum_{q=1}^{c-1} SL_q(l_{i,j}) \quad (10)$$

$$SL_q(l_{i,j}) \leq RC_q(l_{i,j}) - min(S_q(l_{i,j}), RC_q(l_{i,j}) \quad (11)$$

Step 4 (Kicking): Try to kick the assigned resources partially or totally starting from the lowest priority slice upwards through the lower adjacent slice until there are enough resources to satisfy $d_w(CT_c)$. If the sum of squatted higher resources plus the squatted lower resources plus the kicked lower resources satisfy $d_w(CT_c)$, then accept $d_w(CT_c)$ and count the kicked demands as blocked demand for the same slice else, $d_w(CT_c)$ will be rejected. Eq. (12) ensures that the total allocatable resources cannot exceed the total of slice resource constraint plus all squatted resources in both squatting high and low plus all kicked resources from the lower priority slices. Moreover, the total kicked resources from lower slice q, $K_q(l_{i,j})$ cannot exceed the slice resource constraints $RC_q(l_{i,j})$ as Eq. (13). Note that the lowest priority slice cannot use kicking strategy.

$$S_c(l_{i,j}) \leq RC_c(l_{i,j}) + \sum_{q=c+1}^{N} SH_q(l_{i,j}) + \sum_{q=1}^{c-1} SL_q(l_{i,j}) + \sum_{q=1}^{c-1} K_q(l_{i,j}) \quad (12)$$

$$K_q(l_{i,j}) \leq RC_q(l_{i,j}) \quad (13)$$

Obtaining an optimal solution for the above-formulated problem would involve computing all possible paths between source and destination, then enumerating all service deployment combinations in order to identify the optimal solution from all the feasible solutions. Evidently, this is a typical NP-hard problem. As such, exact solutions, as well as approaches based on conventional solvers such as CPLEX and Gurobi to solve the above problem, are not feasible in terms of execution time for delay-sensitive 5G applications which is the target of this paper, especially for large scale networks. Therefore, this motivates the adoption of our heuristic approach as it is able to realize the near-optimal solution with feasible execution time.

V. DEPLOYMENT POLICY OF MULTIPLE NETWORK SLICES

In this section, we discuss the proposed algorithm for serving of priority requests in a multi-slice network. Specifically, we give a detailed discussion of the different steps involved in implementing the algorithm.

Network slicing requires effective QoS management models to provide fast and dynamic detection and reservation of network resources that often vary in type, implementation and priorities. Consequently, the main goal of the deployment process is to optimize the use of resources by effectively allocating different priority service requirements in terms of link resources across the entire network. We use SKM strategy in nodes to optimally assign the demands in terms of bandwidth in a multi-slice network. Since this is an NP-hard problem, even the algorithms proposed would consume a considerable processing load to calculate the solution. As such, they will be forwarded to the NFV architecture under the shape of a Service Chain of VNFs. With NFV, each service instance is composed of a sequence of virtual network nodes (VNNs) and virtual links, which can be illustrated as a service chain (SC). VNNs which carry dedicated VNFs can be deployed onto network data centers (DCs) and run on general-purpose hardware. A virtual link between VNNs can be realized as a multi-hop physical path. Hence, the network slicing resource allocation can be defined as a possible path that slice traffic should follow in infrastructure networks with adequate resource availability. To this end, SKM is used in nodes along the requested routed physical path to organize VNFs execution in the reconfigurable graphs (i.e., VNF forwarding graphs (VNF-FGs) or as defined by ETSI SC) in order to realize network service elastically as shown in Fig. 2. This is because the traffic flowing through VNF-FGs can exhibit high throughput and high burstiness in the dynamic nature of bandwidth-intensive services.

The deployment process includes three main steps to allocate each demand: 1) Search for all possible paths from source to destination with the specified routing algorithm; 2) Allocation decisions and 3) Optimal path selection strategy. Below is a detailed description of the steps.

A. ROUTING ALGORITHM STEP

To find all possible paths to allocate the service request from the source node (s) to the destination node (r),

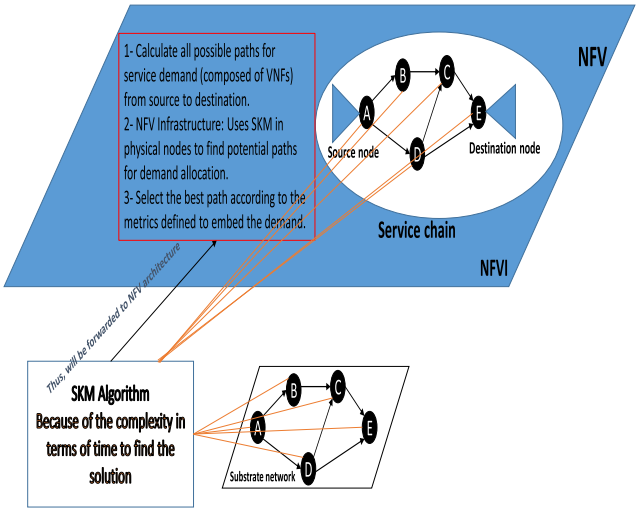


FIGURE 2. An illustrative diagram showing how SKM organizes the execution of SFCVNFs on a shared underlying network to allocate demands.

several algorithms can be used like brute force and others. Because of the path computation complexity, in this study, we are adopting the k-shortest path (KSP) algorithm. Moreover, in this study, all possible paths are examined first, and then the best routing path is determined by examining node per node to see if there are enough resources. This is performed using SKM according to service requests and the available resources in the network.

B. ALLOCATION DECISIONS STEP

Since the substrate is shared, the number of concurrent instances can be optimized through approaches and algorithms to control how the resources are accessed and dynamically optimize the network utilization according to a set of service slices. The allocation decisions steps are described below:

- Check all potential paths according to the available resources' metric defined using a specific allocation strategy.
- Adopt a specific allocation strategy (SKM) for individual node allocation optimization based on efficient utilization. SKM performs the admission control to check if the resources of the user demands are sufficient for the QoS requirements.
- For each path (check node-per-node), calculate the available resources along the routing path by using a specific allocation strategy according to the demand requirements. Note that if there are not enough resources in one link to allocate the demand, the path will be discarded.

Algorithm 1 summarizes the steps of SKM to allocate the demand in a multi-slice network.

C. PATH SELECTION STRATEGY STEP

After checking all feasible routing paths by using SKM in nodes, we must choose the best path based on available resources in the links belonging to that path in order to achieve reliable user traffic. In this work, allocating a demand in the optimal path depends on three tasks: 1) SKM calculates the resources available in the links along the routing paths. Up to this task, SKM gets an overview of the link capacity and the resources available in the link. Then, 2) determines the highest available resource path, taking into account service quality constraints. Finally, in 3) in the case of the presence of two or more paths, having the same available resources, the demand will be allocated in the path with the least bandwidth resource consumption.

Algorithm 1 SKM Algorithm

Input: Set of all Kshortest paths $P_{set}^{s,r}$, demand $d_w = d_w(CT_c)$ to be allocated
Output: Set of accepted paths A_p
Process Assignment;
Initialize Allocation status, Z(d_w): Succeed, R: Reject;
Initialize A_p as empty set;
for *Each* $P_{s,r}^{k} \in P_{set}^{s,r}$ **do**
  for *Each* $l_{i,j} \in P_{s,r}^{k}$ **do**
    if $d_w(CT_c) \leq$ *slice c constraints* $CT_c(l_{i,j})$ **then**
      Execute MAM strategy using Eq. (6) ;
    end
    else if *the total allocatable resources in slice c* $S_c(l_{i,j}) \leq$ *the sum of* $CT_c(l_{i,j})$ *and all squatted resources from higher priority slices* **then**
      Execute RDM strategy using Eq. (8);
    end
    else if $S_c(l_{i,j}) \leq$ *the sum of* $CT_c(l_{i,j})$ *and all squatted resources in both higher and lower priority slices* **then**
      Execute Squatting-Low strategy using Eq. (10);
    end
    else
      found-kick=false;
      for *all slices priority* $<$ *slice c priority* **do**
        if $S_c(l_{i,j}) > CT_c(l_{i,j})$ *plus all squatted resources in both squatting high and low* **then**
          Execute Kicking strategy using Eq. (12)
          found-kick=true;
        end
      end
    end
    if ¬*(found-kick)* **then**
      R: Reject d_w for $P_{s,r}^{n}$;
    end
    $Z(d_w)$: Succeed d_w for $P_{s,r}^{k}$;
    Add $P_{s,r}^{k}$ into A_p as potential path;
  end
end

For task 1, the available link resources at any time can be calculated as below:

$$R_a^t(l_{i,j}) = R(l_{i,j}) - R_z^t(l_{i,j}) \tag{14}$$

$$R_z^t(l_{i,j}) = \sum_{c=1}^{N} R(l_{i,j}) - (RC_c(l_{i,j}) - min(S_c(l_{i,j}), RC_c(l_{i,j}))) \tag{15}$$

Eq. (14) shows the calculation of available resources in a link. Moreover, $R_z^t(l_{i,j})$ can be determined by the summation of the difference between $R(l_{i,j})$, and the minimum between assigned and reserved resources for each slice as in Eq. (15), where N is the number of slices along the link $l_{i,j}$.

Then, the link's available resources is the minimum value of $R_a^t(l_{i,j}) = Min(R_a^t(l_{i,j}))$.

Task 2, aims at selecting the best path taking into account the constraints of bandwidth resources in the links. The best path in relation to QoS constraints can be determined when their links satisfy the resource constraints.

The links meeting the resource constraints are defined by Eq. (16):

$$Max\left\{Min(R_a^t(l_{i,j})) \geq P_{s,r}^k d_w(CT_c)\right\}, \quad \forall l_{i,j} \in P_{s,r}^k,\ k \in [1, K] \tag{16}$$

In the final task, if there are two or more paths with the same amount of optimal available resources, then, select the path with the least bandwidth resource consumption to be optimal as as expressed in Eq. (17).

$$Min\left\{\sum_{\forall l_{i,j} \in P_{s,r}^k} R_z^t(l_{i,j})\right\}, \quad k \in [1, K] \tag{17}$$

Algorithm summarizes the procedure for highest available routing path selection. After defining the optimal path that meets QoS constraints, the demand will be allocated based on the SKM in the network.

Algorithm Path Selector Algorithm

Input: Set of accepted paths A_p
Output: Path connecting a source s to a destination r with highest available resources path $P_{s,r}^k$.
PROCESS
for *each link* $l_{i,j} \in$ *path* $P_{s,r}^k \in A_p$ **do**
  Calculate available resources of a link $l_{i,j}$;
  $R_z^t(l_{i,j}) = \sum_{c=1}^{N}$
  $R(l_{i,j}) - (RC_c(l_{i,j}) - min(S_c(l_{i,j}), RC_c(l_{i,j})))$;
  Determine path available resources;
  $R_a^t(l_{i,j}) \Leftarrow \text{Min}(R_a^t(l_{i,j}))$;
end
for *each* $(P_{s,r}^k \in A_p$ *and* $R_a^t(l_{i,j}) > 0)$ **do**
  Select the optimal path based on highest available resources;
  $Max\left\{Min(R_a^t(l_{i,j})) \geq P_{s,r}^k d_w(CT_c)\right\}, \forall\ l_{i,j} \in P_{s,r}^k, k \in$ [1, K];
  if *two paths or more have same amount of available resources along the path* **then**
    compute the less consumption path of bandwidth resources;
    $Min\left\{\sum_{\forall l_{i,j} \in P_{s,r}^k} R_z^t(l_{i,j})\right\}, k \in [1, K]$;
  end
end

VI. PERFORMANCE EVALUATION

In this section, technical comparison of SKM against the state-of-the-art algorithms, the evaluation methodology for our service deployment policy behavior for both online and offline modes including the performance metrics and description of the simulations scenarios are presented. Later on, the results obtained are presented and discussed.

A. COMPARED ALGORITHMS

BAMs are of great value in the context of efficient and customized use of network resources among several traffic classes (slices). Therefore, in this work, we compare our proposed algorithm against other states of art BAMs. We complement the case study presented by simulating our proposed algorithm using full network topology and comparing the results against the most referenced MAM, RDM, G-RDM and AllocTC. The resource allocation algorithms that are compared in different simulations are summarized in Table 2 with their key attributes indicated. The algorithms are compared considering a number of simulation scenarios with each scenario intended to meet a given objective. The scenarios considered for the performance analysis are indicated in VI-D. In all simulations, algorithms were developed using Eclipse IDE for Java Developers, version: Mars.2 Release (4.5.2) and conducted on a desktop computer running Windows operating system with the following specifications: Intel(R) Core(TM) 2 CPU 6400 @ 2.13GHz Memory 6GB.

B. OFFLINE AND ONLINE BEHAVIORS OF THE PROPOSED DEPLOYMENT POLICY

The deployment policy proposed in this paper is designed to customize service demands between a set of network slices to work with both offline and online scenarios. In offline mode, all demands are known in advance, while in the online scenario, they are assumed to arrive in real-time, where each demand has a lifetime. The following sub-subsections introduce the overall idea of each scenario.

1) OFFLINE BEHAVIOR OF THE PROPOSED DEPLOYMENT POLICY

Fig. 3 is a flowchart showing the general procedures of the offline behavior of the proposed deployment policy. This behavior includes three phases as follows:

1) Initialization and routing paths to find all possible paths;
2) Then, the allocation decisions (SKM) phase;

TABLE 2. Summary of the main attributes of the comparison algorithms.

Algorithm	Key attributes
MAM [33]	I) It is a strict allocation model of the link resources. Each $CT_c(l_{i,j})$ has its private resources, and if the latter is not used, it cannot be allocated to another $CT_c(l_{i,j})$. II) It gives poor use of resources and there is no guarantee of high acceptance of higher priority slices under congested scenarios. III) Not support for preemption action.
RDM[34]	I) It is a nested allocation model of the link resources. The highest $CTs(l_{i,j})$ priority can reuse the free resources of lower priority $CTs(l_{i,j})$. Therefore, the reservation is made from top to bottom and not the reverse. II) It offers low resource usage but better than MAM and there is no guarantee of high acceptance of higher priority slices under congested scenarios. III) Supports higher priority class to preempt lower ones.
G-BAM[25]	I) It switches autonomously between models (MAM and RDM) based on a controller. II) It offers low resource usage and there is no guarantee of high acceptance of higher priority slices under congested scenarios. III) Supports higher priority slice to preempt lower ones.
AllocTC [35]	I) It allows an opportunistic sharing of the link resources among the different slices. It is regarded as an improvement of the RDM model because it not only allows a top-down but down-top reservation as well. II) It offers high resource usage but there is no guarantee of high acceptance of higher priority slices under congested scenarios. III) Supports lower or higher priority slices to preempt each other.
SKM [23]	I) is a smoother BAM policy transition among existing policy alternatives resulting from MAM, RDM, AllocTC adoption independently in a single solution through squatting strategy. The squatting strategy allows sharing unused resources between all $CTs(l_{i,j})$. II) It offers high resource usage and guarantees of high acceptance of higher priority slices under congested scenarios due to kicking operation. III) Supports higher priority slice to kick lower ones.

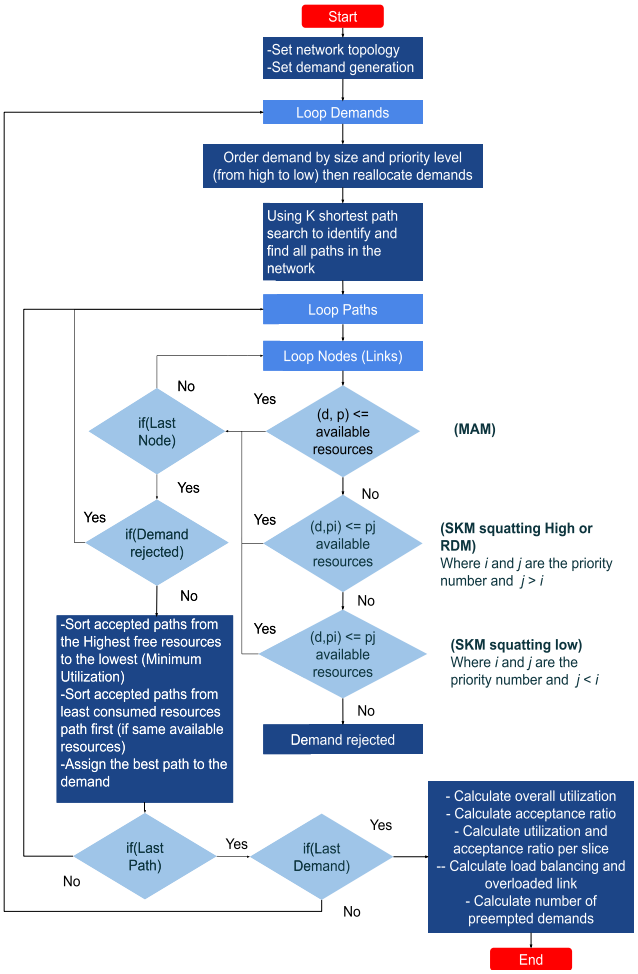


FIGURE 3. Flowchart 1 presenting the general structure of the methodology used in the offline mode of our proposed deployment policy. It starts by the initialization and routing phase, followed by the allocating step, and concludes by the evaluation phase.

3) Finally, the evaluation phase to assess the performance of our proposed service deployment policy including several metrics.

Additionally, this behavior includes a new decision strategy for allocating resources to demands and promotes resource management and reservation. As we mentioned before, in offline behavior, the numbers of demands are known in advance. Therefore, to simplify the path computation in terms of resource allocation, we order the demands according to their priorities and capacities. This means that if two or more demands have the same priority, the largest demand is allocated first to keep resource usage high in most cases. This is meant to simplify the procedure of assigning accepted demands in all links along the routing path since this strategy will make kicking unnecessary (i.e., kicking action will be unnecessary since the higher priorities are processed before).

2) ONLINE BEHAVIOR FOR THE PROPOSED SERVICE DEPLOYMENT POLICY

Fig. 4 is a flowchart showing the general procedures of the online behavior of the proposed deployment policy. By applying this behavior, network traffic can be distributed fairly according to the QoS strategy across all links along the paths. This gives efficient use of network resources and solves online allocation problems such as priority forwarding across all links along paths throughout unit times. In this behavior, the demands are arranged according to size and priority to minimize the number of kicking operations. The contrast between SKM offline and online behavior is that in offline mode, sorting is done before the process of the allocation of Alg 1 in each unit time. In other words, for each unit of time, the algorithm fetches a set of multiple demands sequentially from the demand generation file (D) list and checks for the expiration of the allocated demands. After checking the expiration stage, demands will be ranked according to size and priority level from highest to lowest. Once the arrangement stage occurs, the process assignment of Alg 4 will be used to allocate demands along the network topology paths. Once successful allocating occurs, the algorithm updates all the changed substrate network resources and moves to the

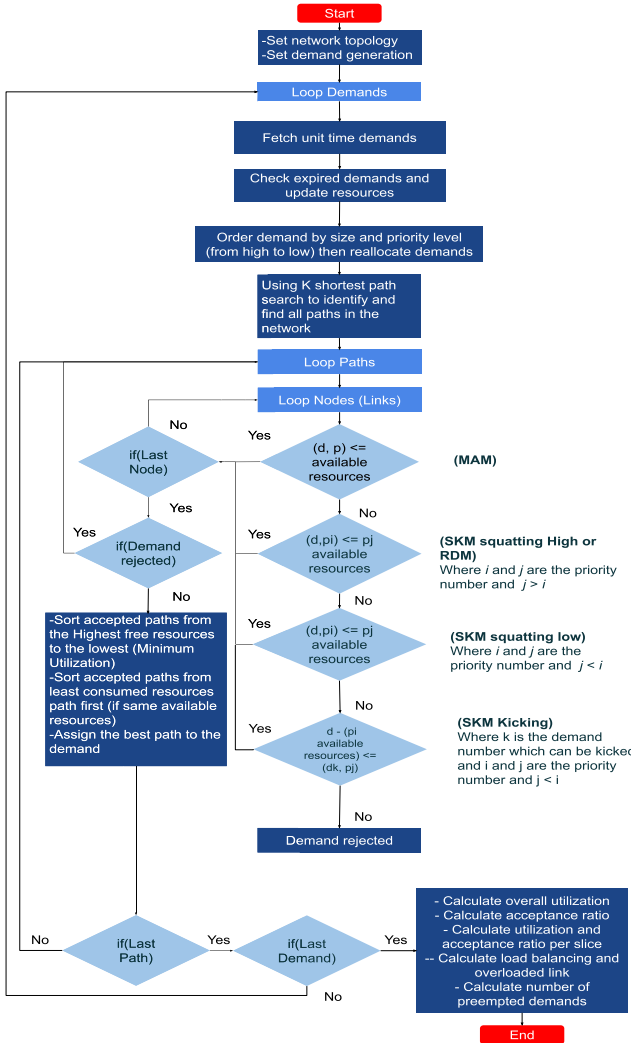


FIGURE 4. **Flowchart 2 presenting the general structure of the methodology used in the online mode of our proposed deployment policy. It starts by the initialization and routing phase, followed by the allocating and updating phase, and concludes by the evaluation phase.**

next unit time. However, if the selected paths from source to destination do not have sufficient resources to accommodate demands at a unit time, the algorithm rejects these demands and moves to the next unit time. This process continues until no more requests are processed. Please note that in both offline/online modes, the demands arranging step improves the resource usage in the network because arranging demands according to the size leads to higher utilization rate in most cases. Moreover, arranging demands, according to the priority, guarantees the least amount of kicking procedure.

C. PERFORMANCE METRICS

The performance of our service deployment policy is compared with the chosen state-of-the-art policies with regards to several performance metrics including the total acceptance ratio (AR) and the overall utilization (U), among others. These are commonly used metrics in the literature for assessing the performance of the resource allocation algorithms [45], [46]. All the performance metrics used are described below for each one of the behaviors.

1) OFFLINE PERFORMANCE METRICS

For the case of permanent demands (without lifetime), the total acceptance ratio (AR), the total blocking probability (Bp), the total utilization (U), the acceptance ratio per slice (ARc), the blocking probability per slice (Bpc), the utilization per slice (Uc), load balancing and overloaded link can be evaluated in Eq. (18-25) as below:

Acceptance ratio, AR:

$$AR = AD/D \tag{18}$$

Acceptance ratio per slice, AR_c:

$$AR_c = AD_c/D_c \tag{19}$$

Blocking probability, Bp:

$$Bp = BD/D \tag{20}$$

Blocking probability per slice, Bp_c:

$$Bp_c = BD_c/D_c \tag{21}$$

Utilization, U:

$$U = \frac{1}{L}\sum_{\forall l_{i,j} \in L} \frac{R_z(l_{i,j})}{R(l_{i,j})} \tag{22}$$

Utilization per slice, U_c:

$$U_c = \frac{1}{L}\sum_{\forall CT_c(l_{i,j}) \in L} \frac{S_c(l_{i,j})}{R(l_{i,j})} \tag{23}$$

Load balancing, $LB(L)$:

$$LB(L) = \frac{\sum_{\forall l_{i,j} \in L}(U-\mu)^2}{|L|} \tag{24}$$

Overloaded link, L_{ov}:

$$L_{ov} = Max(U - \mu), \quad \forall l_{i,j} \in L \tag{25}$$

Please note that, AD, AD_c, BD, BD_c, D, D_c, $s_c(l_{i,j})$, μ and $R(l_{i,j})$ definitions are given in Table 1.

2) ONLINE PERFORMANCE METRICS

The metrics for the finite duration (online) demands considered in our work are as follows:

Average acceptance ratio, $AR(T)$: This parameter is a direct measure of how an algorithm is able to share the resources among the multiple demands in an effective manner. This is expressed as a ratio of the allocated demands to the total demands in the system, whereas, the demands in the system include both the admitted and pending demands. Therefore, the allocating algorithm should guarantee a good AR performance with the constraint that there is no degradation in the QoS of the allocated users. Mathematically, the average AR is given in Eq. (26) [45]. Where the observation time of the system is from t until T.

$$AR(T) = \frac{1}{T}\sum_{\forall t \in T} \frac{AD(T)}{D(T)} \tag{26}$$

Average blocking probability, $Bp(T)$: This parameter is evaluated as the ratio between the total number of blocked demands and the total number of demands received by the system throughout the entire simulation window. Mathematically, the average $Bp(T)$ is given in Eq.(27).

$$Bp(T) = \frac{1}{T} \sum_{\forall t \in T} \frac{BD(T)}{D(T)} \tag{27}$$

Average acceptance ratio per slice, $AR_c(T)$: This parameter is a direct measure of how an algorithm is able to share the resources into set of traffic slices among the multiple demands in an effective manner. This is expressed as a ratio of accepted demands by each slice separately and the total demands for the same slice throughout the entire simulation window. Mathematically, the average $AR_c(T)$ is given in Eq.(28).

$$AR_c(T) = \frac{1}{T} \sum_{\forall t \in T} \frac{AD_c(T)}{D_c(T)} \tag{28}$$

Average blocking probability per slice, $Bp_c(T)$: This parameter is a direct measure of the ratio between the total blocked demands by each slice separately and the total demands for the same slice received by throughout the entire simulation window. Mathematically, the average $Bp_c(T)$ is given in Eq.(29).

$$Bp_c(T) = \frac{1}{T} \sum_{\forall t \in T} \frac{BD_c(T)}{D_c(T)} \tag{29}$$

Average resource utilization, $U(T)$: This parameter considers the average utilization of the links throughout the entire simulation window. The link resource utilization as defined in [46] is the ratio of the used resources to the link capacity averaged over all substrate links. The mathematical formulation of these parameters is given in Eq.(30).

$$U(T) = \frac{1}{T} \sum_{t \in T} \frac{1}{L} \sum_{\forall l_{i,j} \in L} \frac{R_z^t(l_{i,j})}{R(l_{i,j})} \tag{30}$$

Average resource utilization per slice, $U_c(T)$: This parameter takes into account the average use of slices in links throughout the entire simulation window. The utilization per slice is the ratio of the used resources by each slice separately to the total capacity of resources of the same slice. The mathematical formulation of these parameters is given in Eq.(30).

$$U_c(T) = \frac{1}{T} \sum_{t \in T} \frac{1}{L} \sum_{\forall CT_c(l_{i,j}) \in L} \frac{S_c(l_{i,j})}{R(l_{i,j})} \tag{31}$$

Average load balancing, $LB(L)(T)$: In this work, we use the variance of the link resource consumption to calculate the load balancing in the network. Mathematically, the average load balancing performance, LB(L)(T) across the links of the full network is given in Eq.(32) [46].

$$LB(L)(T) = \frac{\sum_{\forall l_{i,j} \in L} (U(T) - \mu)^2}{|L|} \tag{32}$$

where L is the set of all links in the network and $|L|$ is the cardinality of this set. U(T) as illustrated in Eq.(30) is the average resource utilization on the link $l_{i,j}$ and μ is the mean value of this parameter along the network.

Average overloaded link, $L_{ov}(T)$: High overloaded links will be the reason for having long term queues, and thus, higher delay and higher packet loss rate will occur. Moreover, in order to achieve a better QoS for the service request, the link loads and the number of links across the network should be reduced. The performance of overloaded link across the network can be expressed mathematically by Eq.(33).

$$L_{ov}(T) = Max(U(T) - \mu), \quad \forall l_{i,j} \in L \tag{33}$$

3) EXAMPLE OF OUR PROPOSED ONLINE DEPLOYMENT POLICY

In this example: The network topology consists of 6 Nodes and 9 links as illustrated in Fig. 5. In this substrate network, it is assumed that all links have the same capacities which are equal to 30 units. Moreover, each link is split into three priority slices having same amount of resources equal to 10 units. Four requests attempt to be mapped based on the resources available in the network as indicated below. For each request, the deployment algorithm will execute three steps as follows: i) In the routing algorithm step, we assumed that the K-shortest path search algorithm is applied with the value of k set to 2 in order to allocate the request; ii) Then, SKM strategy will be used to allocate the requests across the network, and iii) Finally, the performance of our deployment policy is evaluated through a number of metrics. Moreover, the generation rate is one request per each unit time as follows:

#1: From A to D, 15 units, priority 2, duration = 3
#2: From A to E, 10 units priority 3, duration = 2
#3: From A to F, 20 units priority 3, duration = 4
#4: From A to F, 24 units priority 1, duration = 6

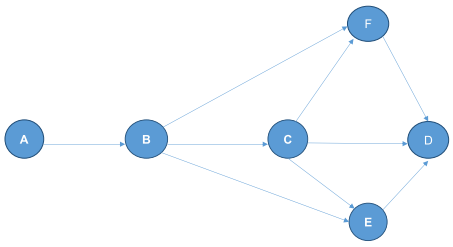


FIGURE 5. An illustration of the substrate network composed of six nodes and nine links in which the above four requests have to be mapped.

Table 3 illustrates the online behavior of the proposed policy in the example shown above, taking into account the online mode in terms of resource allocation and on-demand reservation. This is realized by considering traffic slices and link capacities. Furthermore, the above example shows that SKM can make efficient path setting decisions according to the

TABLE 3. A numerical example illustrating the execution of the proposed algorithm.

#of demand: $d_p(t)$ & path selection	**Allocation (SKM on-line)**						
3 PRIORITY SLICES	**(Unit time 1) Paths to be checked/sorted ($P_{A,B,C,D}, P_{A,B,F,D}$)**						
	Paths to be selected	**Expired demands**	**New demands to be processed**	**Available Resources in each link of path**	**Alive demands after sorting**	**Execution**	**Evaluated Metric**
$\#1:15_2(3)$ **The selected path after checked all paths is $P_{A,B,C,D}$**	$P_{A,B,C,D}$ in this path: [4]nodes and [3] links	–	$\#1:15_2(3)$	A-B (10,10,10) B-C (10,10,10) C-D (10,10,10)	–	(10,0,5) RDM (10,0,5) RDM (10,0,5) RDM	Ra_(A,B) =30-15 = 15 units Ra_(B,C) =30-15 = 15 units Ra_(C,D) =30-15 = 15 units Min available resources of Ra given slice for links along the path (Min Ra) = 15 units. Consuming resources along the path = 15+15+15 = 45 units
	$P_{A,B,F,D}$ in this path: [4]nodes and [3] links			A-B (10,10,10) B-F (10,10,10) F-D (10,10,10)		(10,0,5) RDM (10,0,5) RDM (10,0,5) RDM	Ra_(A,B) =30-15 = 15 units Ra_(B,F) =30-15 = 15 units Ra_(F,D) =30-15 = 15 units Min available resources of Ra given slice for links along the path (Min Ra) = 15 units. Consuming resources along the path = 15+15+15 = 45 units
3 PRIORITY SLICES	**(Unit time 2) Paths to be checked/sorted ($P_{A,B,E}, P_{A,B,C,E}$)**						
	Paths to be selected	**Expired demands**	**New demands to be processed**	**Available Resources in each link of path**	**Alive demands after sorting**	**Execution**	**Evaluated Metric**
$\#2:10_3(2)$ **The selected path after checked all paths is $P_{A,B,E}$ (the least consumed resources path)**	$P_{A,B,E}$ in this path: [3] nodes and [2] links	–	$\#2:10_3(2)$	A-B (10,0,5) B-E (10,10,10)	$\#2:10_3(2)$ $\#1:15_2(2)$	(5,0,0) SL1_(A-B) (10,10,0) MAM	Ra_(A,B) = 5 units Ra_(B,E) = 20 units Min available resources on the other priority slices for links along the path =5 units. Consuming resources along the path = 25+10 = 35 units
	$P_{A,B,C,E}$ in this path: [4] nodes and [3] links			A-B (10,0,5) B-C (10,0,5) C-E (10,10,10)		(5,0,0) SL1_(A-B) (5,0,0) SL1_(B-C) (10,10,0)MAM	Ra_(A,B) = 5 units Ra_(B,C) = 5 units Ra_(C,E) = 20 units Min available resources on the other priority slices for links along the path =5 units. Consuming resources along the path = 25+25+10 = 60 units
3 PRIORITY SLICES	**(Unit time 3) Paths to be checked/sorted ($P_{A,B,F}, P_{A,B,C,F}$)**						
	Paths to be selected	**Expired demands**	**New demands to be processed**	**Available Resources in each link of path**	**Alive demands after sorting**	**Execution**	**Evaluated Metric**
$\#3:20_3(4)$ **The selected path after checked all paths is $P_{A,B,F}$ (the least consumed resources path)**	$P_{A,B,F}$ in this path: [3] nodes and [2] links	–	$\#3:20_3(4)$	A-B (5,0,0) B-F (10,10,10)	$\#3:20_3(4)$ $\#2:10_3(1)$ $\#1:15_2(1)$	(0,0,0) K2_(A-B) (10,0,0) SL2_(B-F)	Ra_(A,B) = 0 units Ra_(B,F) = 10 units Min available resources on the other priority slices for links along the path = 0 units. Consuming resources along the path = 30+20 = 50 units
	$P_{A,B,C,F}$ in this path: [4] nodes and [3] links			A-B (5,0,0) B-C (10,0,5) C-F (10,10,10)		(0,0,0) K2_(A-B) (0,10,0) K2_(B-C) (10,0,0) MAM	Ra_(A,B) = 0 units Ra_(B,C) = 10 units Ra_(C,F) = 10 units Min available resources on the other priority slices for links along the path = 0 units. Consuming resources along the path = 30+20+20 = 70 units
3 PRIORITY SLICES	**(Unit time 4) Paths to be checked/sorted ($P_{A,B,F}, P_{A,B,C,F}$)**						
	Paths to be selected	**Expired demands**	**New demands to be processed**	**Available Resources in each link of path**	**Alive demands after sorting**	**Execution**	**Evaluated Metric**
$\#4:24_1(6)$ **The demand is rejected**	$P_{A,B,F}$ in this path: [3] nodes and [2] links	$\#2:10_3(0)$	$\#4:24_1(6)$	A-B (5,0,5) B-F (10,0,0)	$\#3:20_2(3)$ $\#4:24_1(6)$	(5,0,5) Rejected (10,0,0) Rejected	Discarded path due to rejected allocation
	$P_{A,B,C,F}$ in this path: [4]nodes and [3] links			A-B (5,0,5) B-C (10,10,10) C-F (10,10,10)		(5,0,5) Rejected (0,0,6) RDM (0,0,6) RDM	Discarded path due to rejected allocation

priority of requests. In the table, the first column represents the service requirements and the best paths to set on the substrate network. The second column on the left shows the routing step for the proposed policy. In each time unit, the algorithm first verifies that allocated requests have expired and the substrate network is updated. Then, the algorithm performs the allocation process as illustrated in the third and fifth columns (expired requests, and the resources available at each link of the path). For example, the algorithm in the fourth unit time updates the network resources because the

TABLE 4. Summary of the results after executing the proposed algorithm.

Links Utilization:	Utilization per slice:	Accepted demands per slice:
Utilization for link (A - B) = (20) / 30 = 66.67% Utilization for link (B - C) = (0) / 30 = 0% Utilization for link (C - D) = (0) / 30 = 0% Utilization for link (B - E) = (0) / 30 = 0% Utilization for link (B - F) = (20) / 30 = 66.67% Utilization for link (C - E) = 0% Utilization for link (C - F) = 0% Utilization for link (E - D) = 0% Utilization for link (F - D) = 0%	Utilization for slice (1) = 0 / (9*30) = 0% Utilization for slice (2) = (0) / (9*30) = 0% Utilization for slice (3) = (20) / (9*30) = 7.40%	For slice (1): 0 Demand(s) of 1 - acceptance = 0% For slice (2): 0 Demand(s) of 1 - acceptance = 0% For slice (3): 2 Demand(s) of 2 - acceptance = 100.00%
Average utilization of the Network = (20/30 + 20/30) / 9 = 14.81%		
Average acceptance ratio = 2/ 4 = 50%		
LB(L) = $[(66.67\% - 14.81\%)^2 + (66.67\% - 14.81\%)^2 + (0 - 14.81\%)^2 + (0 - 14.81\%)^2 + (0 - 14.81\%)^2 + (0 - 14.81\%)^2 + (0 - 14.81\%)^2 + (0 - 14.81\%)^2 + (0 - 14.81\%)^2]/ 9 = 0.26$		
L_{ov} = (66.67% - 14.81%) = 0.52		
Number of preempted demands = 1 ($\#1 : 15_{2,3}(3)$)		

request #2 : $10_3(0)$ has expired, and then begins the allocation process for the new arrival request #4 : $24_1(6)$. Moreover, during the allocation process, the algorithm performs a sorting operation for all requests according to size and priority to ensure that the higher priority request is allocated first as indicated in the sixth column (Alive demands after sorting). As an example showing how to implement the sorting operation, in the fourth time unit, the algorithm implemented the sort operation by rearranging all live requests including new access request #4 : $24_1(6)$. Accordingly, the requests #3 : $20_2(3)$ and #4 : $24_1(6)$ are assigned, respectively. In addition, in this example, in the third unit time, SKM is executing the kicking operation because there aren't enough resources in the network to set the new arrival request #3 : $20_3(3)$ which has the highest priority. This is accomplished by verifying all of the least-priority requests that are not expired that can be expelled. Thus, the request #1 : $15_2(3)$ is expelled from the substrate network to assign the request #3 : $20_3(3)$ as indicated in the seventh column (Execution). In the last column on the right, the algorithm determines which routing path that can be optimized for assigning requests based on available resources. The results of the evaluation metrics are shown in the Table 4, which reflects the online behavior performance of the proposed algorithm in terms of $U_c(T)$, U(T), $AR_c(T)$, Bp(T), AR(T), LB(L)(T) and $L_ov(T)$. From the shown results, the highest priority slice (Slice 3), assigned two requests during observation time slice 3 #2 : $10_3(2)$ and #3 : $20_3(4)$ along the substrate network.

D. EVALUATION SCENARIOS

We carried out our simulations scenarios to fully demonstrate the difference in the performance between the SKM and the BAMs. It is essential to mention that the potential dynamic behavior of our proposed deployment algorithm is the target of the presented simulations. These simulations focus on validating the reproducibility characteristics of our algorithm to ensure the QoS levels (especially the higher priority slices) and to achieve high resource utilization. Moreover, five sets of simulation scenarios, aiming at evaluating our proposed algorithm performance, are conducted in this paper:

1) Scenario one: We generally evaluate our proposed algorithm performance in terms of U(T), $U_c(T)$, AR(T), $AR_c(T)$ and P_{re} in full network by comparing our solution against the most referenced models, MAM, RDM and G-RDM in one scenario similar to [25], as explained in VI-E. The objective of this scenario is to validate the techniques of bandwidth allocation approach of SKM and their ability to generate high admission for the higher priority slices across full network.
2) In the remaining sets of the simulation scenarios, we investigate our proposed solution performance on limited resource networks under different traffic loads. Moreover, this comparison is in terms of U(T), $U_c(T)$, AR(T), $AR_c(T)$ and P_{re}, LB and L_{ov} considering MAM, RDM and AllocTC. We performed our scenarios in both online and offline modes as follows:

- Scenario two: under this scenario, the objective is to evaluate the impact of mesh topology where nodes are reachable in a single hop from each other on the performance of SKM against other algorithms considering different load distributions. An online simulation under mesh network topology and different generated traffic loads for traffic slices of all priorities are considered and described in VI-F. The reason for using the mesh network topology is that bottlenecks are minimal, which gives a more accurate view regarding the scalability of SKM with the size of the network compared to other topologies which have huge bottlenecks links, and this was the basis for this scenario.
- Scenario three: An offline simulation under mesh network topology and different generated traffic loads for traffic slices of all priorities are considered and described in VI-G. The objective of this scenario is to analyze the robustness of SKM under permanent loading stress in a mesh network topology.
- Scenario four: Online simulation under NSF network topology and different generated traffic loads are considered and described in VI-H. The objective of this scenario is to analyze the impact of different network complexity on our proposed algorithm performance

against the other algorithms. The NSF topology has more bottlenecks which further complicates resource allocation and QoS management compared to mesh topology.

- Scenario five: An Offline simulation under NSF network topology and various generated traffic load for traffic slices of all priorities are considered and described in VI-I. The objective of this scenario is to analyze the robustness of SKM under permanent loading stress in the NSF topology.

E. SCENARIO 1: OVERALL PERFORMANCE IN A FULL NETWORK TOPOLOGY

In this evaluation, our proposed solution is compared to G-BAM, MAM and RDM under saturation case. In this scenario, the demands arrive dynamically, and the traffic load is generated high in the lower priority slices to evaluate the performance of each algorithm before and after the saturation case based on different metrics. Our proposed algorithm is specially designed for highly crowded scenarios with strict constraints for the higher priority slices. On the other hand, when the traffic is not saturated, the SKM behaves similar to MAM, RDM and AllocTC in a single solution.

1) SIMULATION SCENARIO SETTINGS

We adopted the settings of [25] where the full network is used to assess the performance of the algorithms. In other words, the choice of this work was deliberate for two main reasons: This work used the full network to assess the performance of the algorithm which proved to be comparable, and in most cases, superior, to state-of-the-art in several performance metrics. The strength of the G-BAM algorithm is derived from the fact that it switches autonomously between models (MAM and RDM) based on a controller. This is performed to decide and/or follow the most adequate transitions according to the defined high-level management configuration requirements such as SLA, QoS, among others. Moreover, this algorithm achieves improvement in network quality parameters like link utilization and preemption. The network topology used is the NTT network containing 55 nodes and 144 links of 622 Mbps (STM-4 - SDH) (see [25]). In this evaluation scenario, a single traffic source (node 0) is defined for each traffic slice (class) and destiny. The destination nodes (54, 52, 48 and 45) are chosen randomly and each link consists of three traffic slices. Slice 3 has the highest priority applications, slice 2 has the medium priority applications, and slice 1 has the lowest priority applications. Nodes (routed paths) are statically defined in order to compete in a high number of links. Consequently, saturated links are forced during simulation in order to observe the consequences:

- **0**->2->5->11->14->19->20->21->22->26->27->35->42->51->**52**
- **0**->2->5->11->14->19->20->21->22->26->27->35->42->51->52->**54**
- **0**->2->5->11->14->19->20->21->22->26->27->35->42->41->**45**
- **0**->2->5->11->14->19->20->21->22->26->27->35->42->41->45->**48**

The configuration parameters of the validation scenario can be summarized as follows:

- Link: 622 Mbps (STM-4 - SDH)
- Existing Traffic slices (classes): slice 1 (CT1), slice 2 (CT2) and slice 3 (CT3).
- Table 5 shows the CTs that can be used through the bandwidth constraint of each slice and obtained in the form of percentage and amount of resources.
- Demand bandwidth is a uniformly distributed bandwidth: 5 Mbps to 15 Mbps.
- Exponential modeled demand request arrival intervals in phases as in Table 6
- Exponentially modeled demand time life: average of 200 seconds (should cause link saturation)
- Simulation stop criteria: 1 h (3600 seconds).

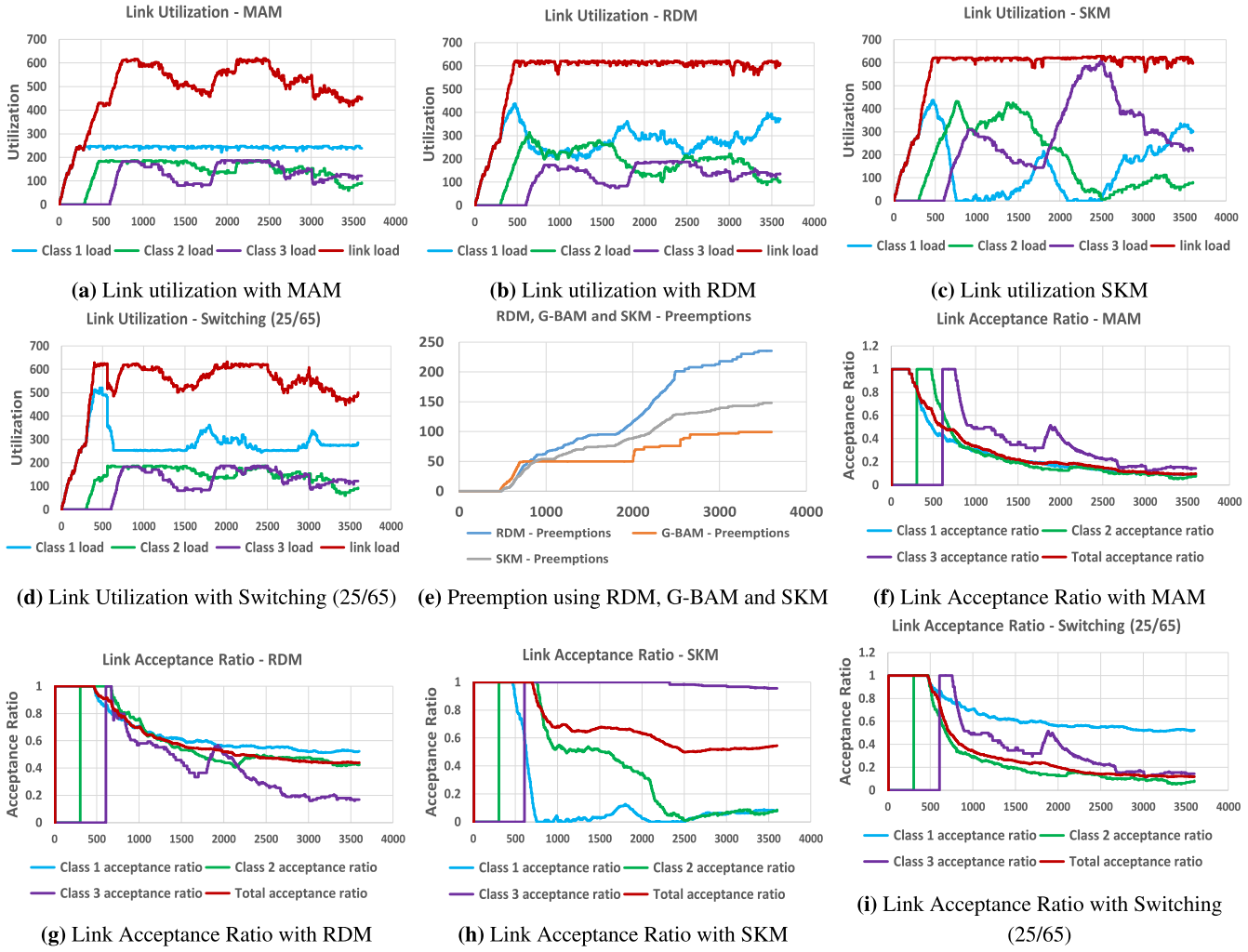
Please note that if RDM algorithm is used in the simulation, the resource constraints for CT3 would be equal to 40% of the link capacity, resource constraints for CT2 would be equal to 70% and resource constraints for CT1 would be equal to 100%. However, if SKM and MAM algorithms are used, the resource constraints for CT3 would be equal to 30% of the link capacity, resource constraints for CT2 would be equal to 30% and resource constraints for CT1 would be equal to 40% as shown in Table 5. Table 6 shows the phases of demand arrival. Phases 1 to 3 create a traffic profile where there are, initially, only low priority demands. These are followed by medium priority demands and then, followed by high priority demands in a high flow rate forcing them to be used to the maximum. In phase 4, the rate of high priority demand arrivals is reduced and, in phase 5, the medium priority ones are reduced. In phase 6, we maintain a low arrival rate of medium priority demands and we increase the arrival rate of high priority demand. In phase 7, we generate a high number of demands for all slices in order to saturate the link. Finally, in phase 8, we reduce the arrivals of high and medium priority

TABLE 5. Bandwidth Constraints (BCs) per CTs.

BC	Max BC %	Max BC (Mbps)	CT per BC	Max BC (%)	Max BC (Mbps)	CT per BC
BC_1	100	622	$CT_1 + CT_2 + CT_3$	40	248.8	CT1
BC_2	70	435.4	$CT_2 + CT_3$	30	186.6	CT2
BC_3	40	248.8	CT_3	30	186.6	CT3

TABLE 6. Rate of demand arrivals by traffic slices (CTs).

Phase	1	2	3	4	5	6	7	8
Time (seconds)	300	600	900	1500	1800	2100	2500	3600
CT1	8	8	8	8	8	8	8	8
CT2	0	8	8	8	100	100	8	50
CT3	0	0	8	100	100	8	8	50



(a) Link utilization with MAM
(b) Link utilization with RDM
(c) Link utilization SKM
(d) Link Utilization with Switching (25/65)
(e) Preemption using RDM, G-BAM and SKM
(f) Link Acceptance Ratio with MAM
(g) Link Acceptance Ratio with RDM
(h) Link Acceptance Ratio with SKM
(i) Link Acceptance Ratio with Switching (25/65)

FIGURE 6. **Comparison of utilization, preemption and acceptance ratio in first scenario.**

demands, and we maintain the high arrival flux of low priority demands.

2) RESULTS EVALUATION

Fig. 6 shows the results obtained by each model in terms of U(T), AR(T) and P_{re}. As shown in Figs. 6a - 6c, the MAM behavior in which there are no preemptions (inherent behavior of MAM) limits the link utilization to 491.35 Mbps on average for the entire simulation window. This results from the fact that, in the simulation, the U(T) most of the time is below the 622 Mbps link capability even when CT1, CT2 and CT3 are congested. This is because MAM does not support resource sharing between slices. At the same time, the simulations use different times of demand arrival phases that increase traffic load in some slices and decrease it in others along the entire simulation window. As such, RDM and SKM show improvement in link utilization as it reaches the maximum capability of 622 Mbps most of the time in relation to MAM. This is because RDM and SKM behaviors allow lower priority slices to share unused resources from higher slices.

Fig. 6d illustrates the behavior of the G-BAM algorithm when RDM switches to MAM after reaching 25 preempted demands and also when MAM switches to RDM after reaching 65% link utilization. This figure represents an example of the G-BAM that uses a controller to manage the link utilization to decrease the high number of preempted demands by using RDM approach alone. However, by using the G-BAM approach, the link utilization will be less than that of RDM and SKM under any cases of traffic load due to switching between MAM and RDM behaviors as illustrated in this example where the U was 520 Mbps.

SKM outperforms MAM, RDM and G-BAM in the highest priority slice by 22.5%, 21.9% and 20.8% in terms of average U_3 respectively due to kicking operation. Similarly, SKM outperforms MAM, RDM and G-BAM by 54.2%, 50.5% and 53.6% respectively in terms of average AR_3 (see Figs. 6f - 6i). Moreover, SKM and RDM have similar performance in terms of AR by achieving 65.4% on average and better than both MAM by 34.7% and G-BAM by 30.8% for traffic patterns in which lower priority slices have greater demands for resources.

Figs. 6e illustrates that, under link saturation, SKM allows the optimization of link utilization with a fewer number of preempted demands (equal to 150 demands) which cannot be achieved by using the RDM approach (approximately 248 demands). This is because the load is low in higher priority slices, so, a smaller number of lower priority demands are kicked. However, if G-BAM behavior is used, the number of preempted is the lowest (105 demands) compared to RDM and SKM, but the link utilization is not improved with the algorithm converted from RDM to MAM.

F. SCENARIO 2: PERFORMANCE IN ONLINE MODE UNDER MESH TOPOLOGY

In this online scenario, we investigate SKM performance on limited resources of mesh network under different traffic loads and under fixed demands lifetime, in terms of several metrics, compared to MAM, RDM and AllocTC. Moreover, this scenario consists of three experiments. The main objective of the experiments below is to analyze the performance of SKM under different load distributions between different priority slices across an entire network. The assessment experiments are as follows:

- Experiment 1: more traffic load in lower priority slices.
- Experiment 2: same traffic load in all priority slices.
- Experiment 3: more traffic load in higher priority slices.

The purpose of experiment one is to demonstrate that SKM has an equivalent behavior to RDM and AllocTC at high loads for lower priority slices along the network. The simulation experiment enforces the sharing or squatting strategy inherent in RDM along the network. The purpose of experiment two is to demonstrate that SKM ensures acceptance of more demands for higher priority slices than AllocTC, RDM and MAM in case of similar loads in traffic slices along the network. The purpose of experiment three is to demonstrate that SKM has an equivalent behavior to AllocTC before the saturation case when the load is high for higher priority slices along the network. This is checked by executing the share strategy of AllocTC or squatting strategy. Moreover, SKM admits more requests than AllocTC and RDM at high loads for higher priority slices, which is due to it being stricter on priorities than the other algorithms after saturation case.

1) SIMULATION SCENARIO SETTINGS

In order to evaluate our solution, the simulated scenario uses different traffic sources and different destinations on the mesh network consisting of 5 nodes and 10 links as shown in Fig. 7. The capacity of the links is equal to $R(l_{i,j}) = 150$ units. Moreover, the link resources are divided into three slices (eMBB, MIoT and uRLLC); each slice has $RC_c(l_{i,j}) = 50$ units. For the routing step, using the k-shortest path, the maximum value of k was set to 5. In all experiments of this simulation scenario, the demands are generated with a fixed lifetime equals to 1-time slot and the size of each demand is also fixed to 1 unit as the minimum granularity for allocation. Each demand has single priority generated in a random manner from (1 to 3) with a generation rate of demands per each unit time equals to 2500 demand. The total number of demands among slices generated until 10 unit time is 25,000 for each experiment.

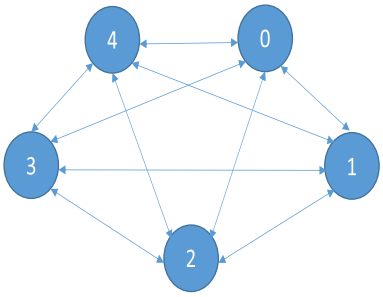


FIGURE 7. Mesh network topology.

2) RESULTS EVALUATION

Table 7 shows the traffic load consideration (number of demanded resources in each slice) for the validation experiments in each slice in each unit time. Please note that in all experiments, the capacity of each slice along the network is 500 unit ($RC_c(l_{i,j})$ * 10 links = total size of the slice across the network).

Figs. 8- 9 show the results of each algorithm in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov} using different traffic load according to experiments 1–3.

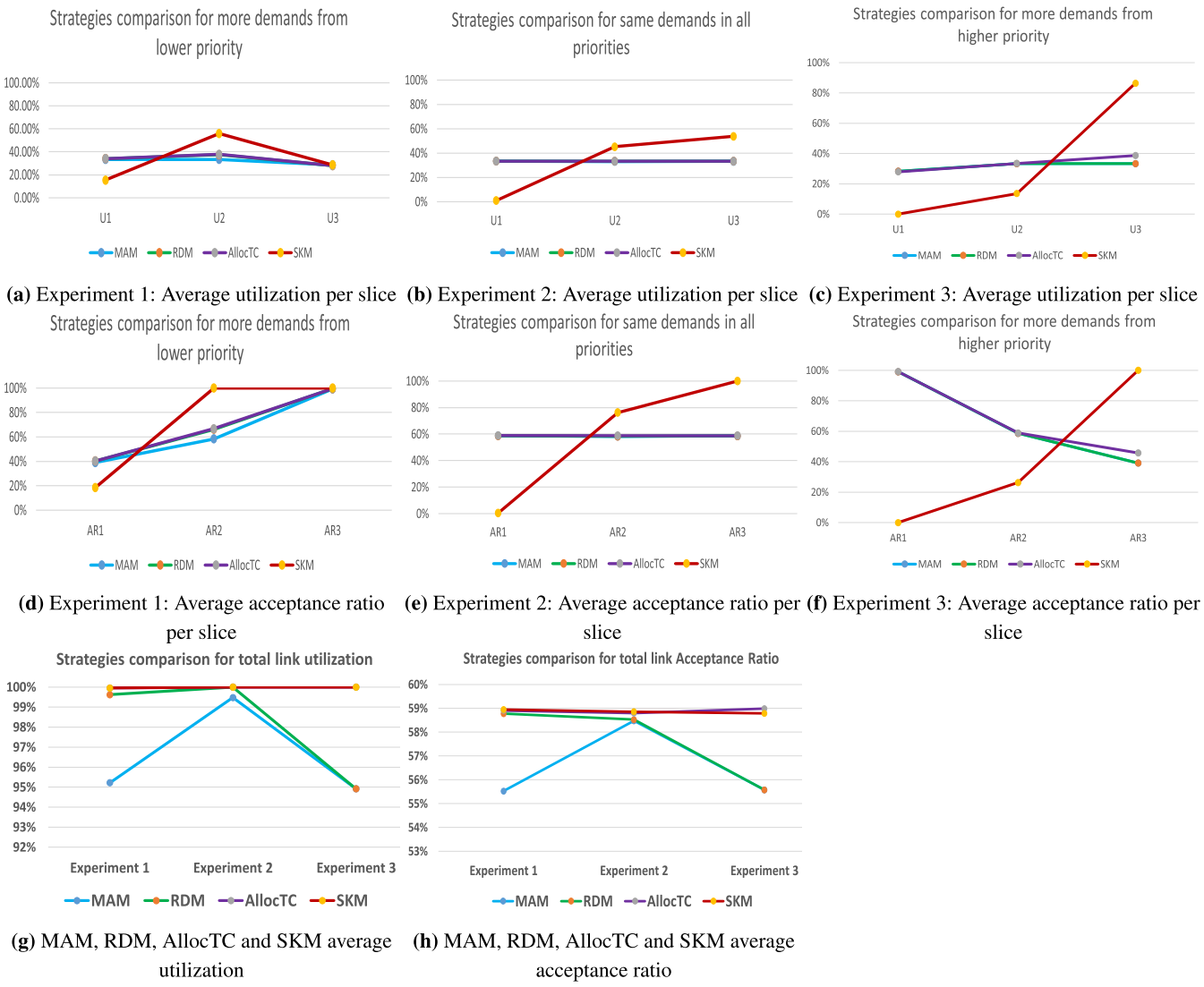
In terms of U and AR, Fig. 8g and Fig. 8h illustrate the results from the three experiments for the MAM, RDM, AllocTC and SKM. **In the first experiment**, SKM, AllocTC and RDM result in 100% U and 58.95% AR where 1475 demands are accepted from 2500 demands per each unit time. On the other hand, MAM achieved the lowest performance and resulted in 95.2% U and 56% AR where 1400 demands are accepted from 2500 demands per each unit time. **In the second experiment**, SKM, AllocTC, RDM and MAM resulted in 100% U and 58.88% AR where 1472 demand from 2500 are accepted per each unit time. **In the third experiment**, SKM and AllocTC have similar performance in terms of U and AR by achieving 100% U and 59% where 1475 demand are accepted from 2500 demand per each unit time. On the other hand, MAM and RDM performance is the lowest one among the four strategies by achieving 94.5% in terms of U and 55.54% in terms of AR. This is because there is no ability to share resources among the slices.

a: CONSIDERING HIGH LOAD IN LOWER PRIORITY SLICES

As expected, SKM, AllocTC, RDM and MAM have similar behavior in terms of U3 and AR3 by achieving 25.60% and 100% (417/417) AR3, respectively. This is because the load distributions on slice 3 across the network was lower than its capacity (the demanded resources for slice 3 was 417 unit). Moreover, SKM outperforms AllocTC, RDM and

TABLE 7. Simulation experiments for the second scenario.

Number of slices in the generating file per-each unit time	Experiment 1 Load volume Traffic (Number of demands)	Experiment 2 Load volume Traffic (Number of demands)	Experiment 3 Load volume Traffic (Number of demands)
slice-Type 1	1250	833	417
slice-Type 2	833	833	833
slice-Type 3	417	834	1250



(a) Experiment 1: Average utilization per slice **(b)** Experiment 2: Average utilization per slice **(c)** Experiment 3: Average utilization per slice

(d) Experiment 1: Average acceptance ratio per slice **(e)** Experiment 2: Average acceptance ratio per slice **(f)** Experiment 3: Average acceptance ratio per slice

(g) MAM, RDM, AllocTC and SKM average utilization **(h)** MAM, RDM, AllocTC and SKM average acceptance ratio

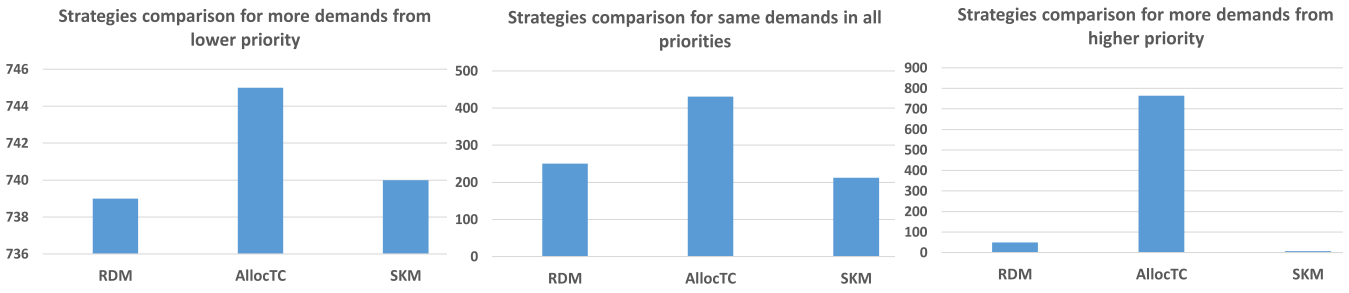
FIGURE 8. Comparison of utilization and acceptance ratio in second scenario.

MAM by 18.02%, 18.34%, 22.54% in terms of U2 due to the kicking operation (see Fig. 8a). Furthermore, SKM, in terms of ARc, achieved 12.5% for slice 2 more than MAM, RDM and AllocTC which achieved 33.34%, 33.76% and 41.58%, respectively (see Fig. 8d).

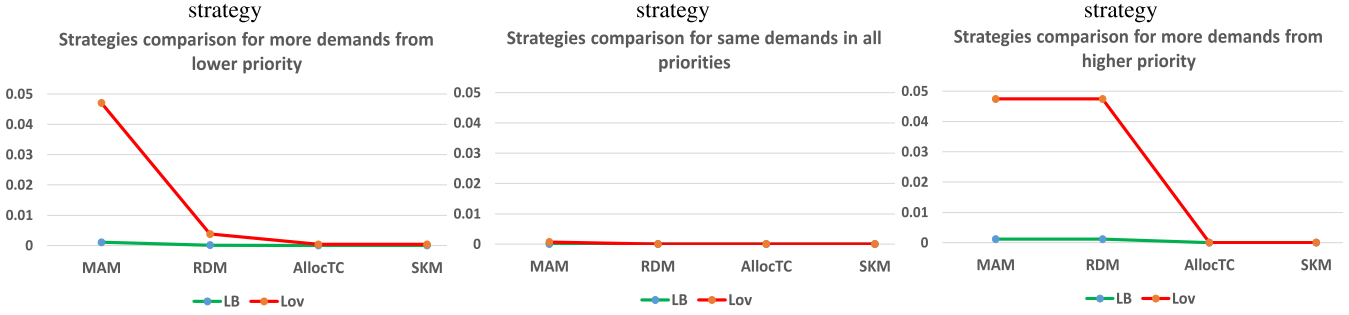
As shown in Fig. 9a, SKM, AllocTC and RDM resulted in 740, 739 and 745 respectively in terms of P_{re}. Moreover, Figs. 9d illustrate that SKM, AllocTC and RDM have a very close performance in terms of LB and L_{ov}. This is because these algorithms have a 100% U value of network resource utilization (almost all links are fully used). Moreover, MAM gives the lowest performance in terms of LB and L_{ov} as it resulted in 0.0011 LB and 0.047 L_{ov} where more links are not fully used across the network.

b: CONSIDERING SAME LOAD IN ALL PRIORITY SLICES

Fig. 8 illustrates that the SKM outperforms MAM, RDM and AllocTC in the highest priority slice by 20.47% in terms of



(a) Experiment 1: preempted demands by each strategy **(b)** Experiment 2: preempted demands by each strategy **(c)** Experiment 3: preempted demands by each strategy



(d) Experiment 1: Load balancing and overloaded link by each strategy **(e)** Experiment 2: Load balancing and overloaded link by each strategy **(f)** Experiment 3: Load balancing and overloaded link by each strategy

FIGURE 9. Comparison of preempted demands, load balancing and overloaded link in second scenario.

TABLE 8. Simulation experiments for the third scenario.

Number of slices in the generating file per-each unit time	Experiment 1 Load volume Traffic (Number of varied demands)	Experiment 2 Load volume Traffic (Number of varied demands)	Experiment 3 Load volume Traffic (Number of varied demands)
slice-Type 1	100, 400, 700, 1001	67, 267, 467, 667	34, 67, 201, 333
slice-Type 2	67, 267, 467, 667	67, 267, 467, 667	67, 267, 467, 667
slice-Type 3	34, 67, 201, 333	67, 267, 467, 667	100, 400, 700, 1001

U3 and 41.17% in terms of AR3. Also, the results show that SKM outperforms MAM, RDM and AllocTC in slice 2 by 11.94% in terms of U2 and by 17.39% in terms of AR2 (as the expected from the behaviors) due to the kicking operation as shown in Fig. 8b and Fig. 8e.

As shown in Fig. 9b, SKM outperforms AllocTC and RDM by 219 and 38 respectively in terms of P_{re} because of the kicking operation. Moreover, Figs. 9e shows that SKM, AllocTC, RDM, and MAM have similar performance in terms of LB and L_{ov} and have resulted in almost zero since all links are used across the network.

c: CONSIDERING HIGH LOAD IN HIGHER PRIORITY SLICES

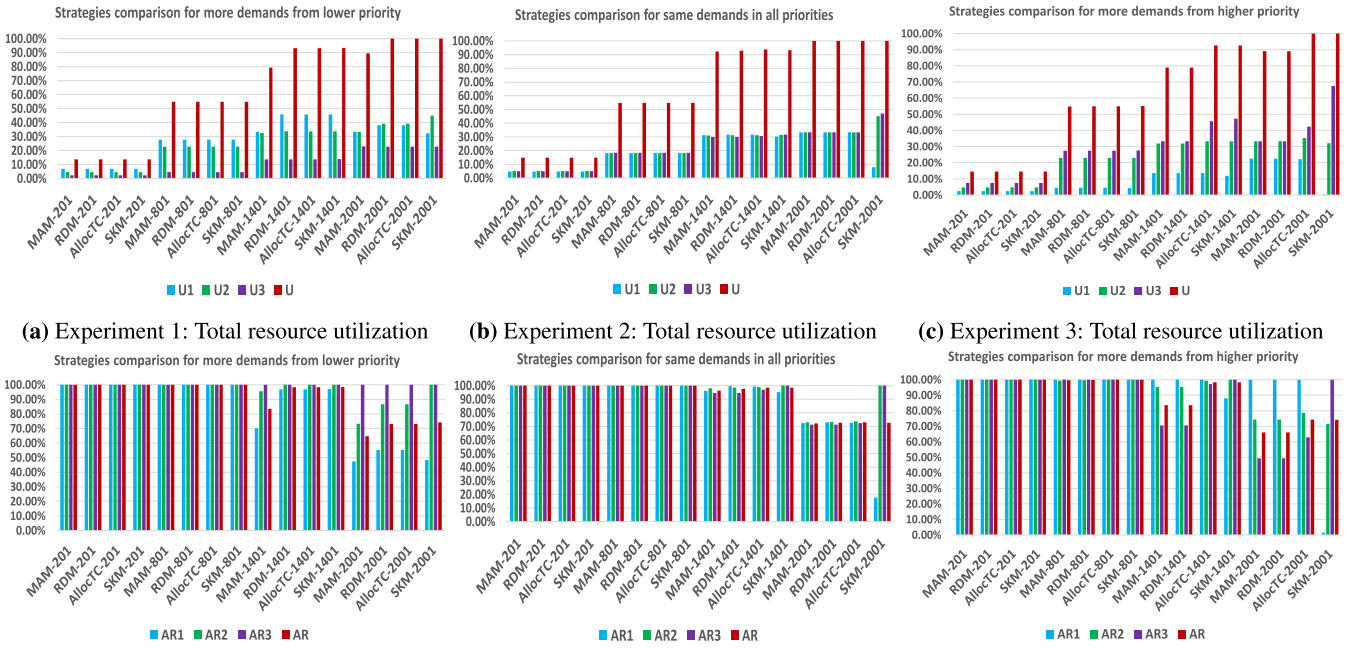
Fig. 8c and Fig. 8f illustrate that the SKM outperforms AllocTC, RDM and MAM in the highest priority slice by 47.79%, 53.14% and 53.14% respectively in terms of U3 and by 54.28%, 60.95%, 60.95% respectively in terms of AR3. Moreover, the results of Fig. 9c shows that SKM outperforms RDM and AllocTC by 657 and 43 respectively in terms of P_{re} since the load was too low in lower slices, so there is no need to use the kicking operation. Further, Fig. 9f illustrates that SKM and AllocTC have a similarly good performance as they achieve zero in terms of both LB and L_{ov} where all links are fully used in the network. Moreover, RDM and MAM give the worst performance in terms of LB and L_{ov} and resulted in 0.0117 and 0.0474, respectively, where more links are not fully used across the network.

G. SCENARIO 3: PERFORMANCE IN OFFLINE MODE UNDER MESH TOPOLOGY

In this offline scenario, we used the same network topology and settings for the second scenario but with infinite demand lifetimes while considering a number of demands that vary from 201 to 2001 for each experiment in the studied scenario (see Table 8). Please note that the goal of this scenario is to investigate the robustness of SKM under constant load stress. Besides, SKM provides a good QoS level among different priority slices.

1) RESULTS EVALUATION

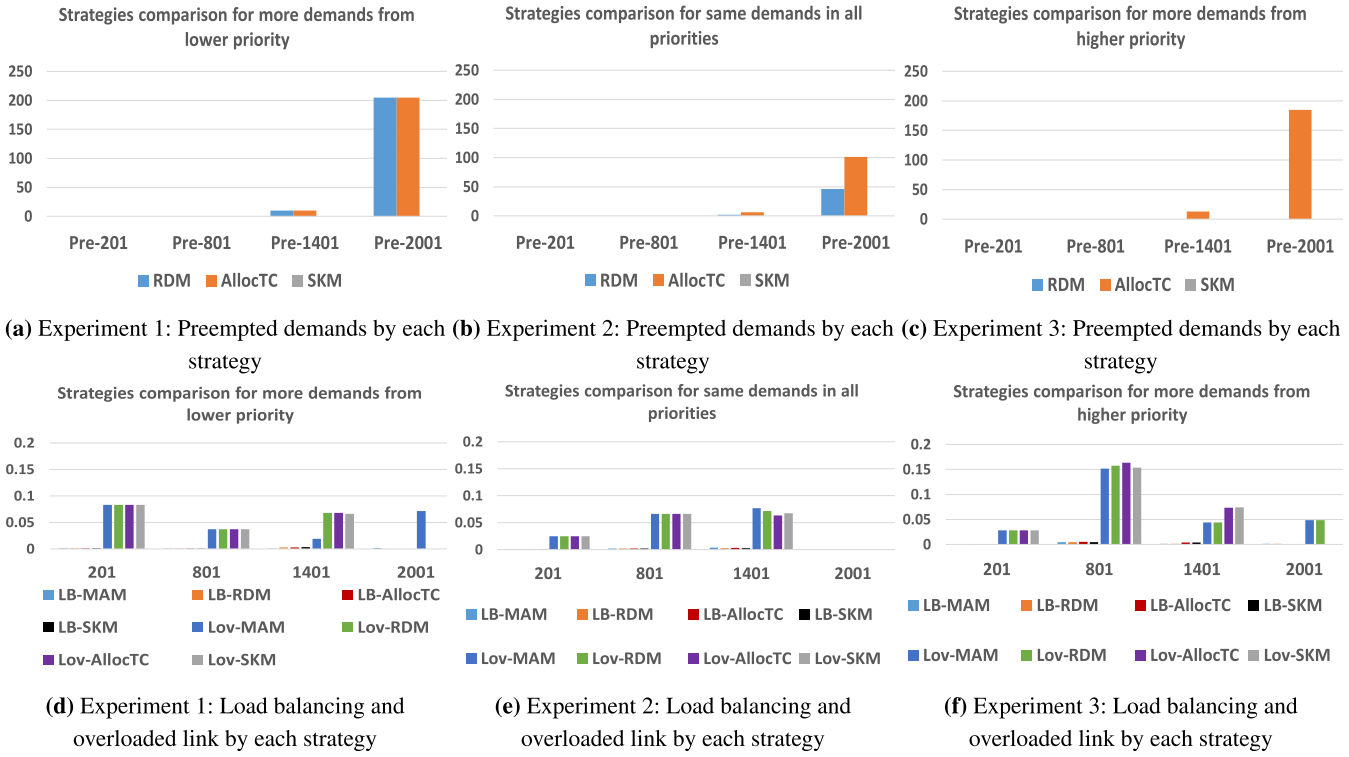
Figs. 10-11 show the results of each algorithm in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov} using different traffic load according to experiments 1–3.



(a) Experiment 1: Total resource utilization **(b)** Experiment 2: Total resource utilization **(c)** Experiment 3: Total resource utilization

(d) Experiment 1: Acceptance ratio **(e)** Experiment 2: Acceptance ratio **(f)** Experiment 3: Acceptance ratio

FIGURE 10. **Comparison of utilization and acceptance ratio in third scenario.**



(a) Experiment 1: Preempted demands by each strategy **(b)** Experiment 2: Preempted demands by each strategy **(c)** Experiment 3: Preempted demands by each strategy

(d) Experiment 1: Load balancing and overloaded link by each strategy **(e)** Experiment 2: Load balancing and overloaded link by each strategy **(f)** Experiment 3: Load balancing and overloaded link by each strategy

FIGURE 11. **Comparison of preempted demands, load balancing and overloaded link in third scenario.**

a: CONSIDERING HIGH LOAD IN LOWER PRIORITY SLICES

Fig. 10a and Fig. 10d reveal that as demand size increases as in the case where the size equals to 2001, all algorithms have similar performance in terms of both U3 and AR3 as they achieve 22.80% and 100% respectively. Moreover, SKM outperforms AllocTC, RDM and MAM by 6.45%, 6.26% and

10.50% in terms of U2 and by 14.44%, 14.44% and 26.84% respectively in terms of AR2 due to the sorting operation of SKM. Furthermore, Figs. 11a- 11c reveal that as demand size increases, irrespective of traffic load distribution for slices, SKM results in superior performance in terms of P_{re} compared to other algorithms as it achieves zero preempted demands due to sorting operation of SKM. Fig. 11d reveals that as demand size increases as in the case where the size equals to 2001, SKM, AllocTC and RDM have similar performance in terms of LB and L_{ov} and result in almost zero since all links are fully used across the network. On the other hand, MAM gives the lowest performance in terms of LB and L_{ov} as it results in 0.0014 and 0.071, respectively where more links are not fully used across the network.

In terms of U and AR, SKM, AllocTC and RDM have similar behavior and result in 100% and 74.10% respectively. This is as expected from the performance of SKM, AllocTC and RDM where the lower slices can share unused resources from the higher ones. On the other hand, MAM gives the lowest performance because there is no sharing of unused resources between different priority slices and results in achieving 89.53% of U and 63.77% of AR.

b: CONSIDERING SAME LOAD IN ALL SLICES

Fig. 10b and Fig. 10e reveal that as demand size increases as in the case where the size is 2001, SKM outperforms the other models in terms of both U3 and AR3 by 12.59% and 27.60%, respectively. Similarly, SKM outperforms the other models in terms of both U2 and AR2 by 11.87% and 26.24% respectively, because of the kicking operation. Moreover, Fig. 11d reveals that as demand size increases as in the case where the size is 2001, all algorithms have similar performance in terms of both LB and L_{ov} with an average value of zero, since all links are fully used across the network. In terms of U and AR, all algorithms have similar performance as they resulted in 100% and 72.96%, respectively. This is because the number of demands was higher than all capacities of slices.

c: CONSIDERING HIGH LOAD IN HIGHER PRIORITY SLICES

Fig. 10c and Fig. 10f reveal that as demand size increases as in the case where the size is 2001, SKM outperforms AllocTC, RDM and MAM in terms of AR3 by 37.16%, 50.75% and 50.75% and by 25.13%, 34.2% and 34.2% respectively in terms of AR3 because of the kicking operation. Moreover, Fig. 11f reveals that as demand size increases as in the case where the size is 2001, SKM and AllocTC have similar performance in terms of LB and L_{ov} and had resulted in zero because all links are fully used across the network. On the other hand, MAM and RDM give the lowest performance in terms of LB and L_{ov} as they result in 0.0011 and 0.049, respectively where more links are not fully used across the network.

In terms of U and AR, SKM and AllocTC have similar behavior as they both resulted in 100% and 74.26% where higher priority slices have greater demands for resources than other slices. This is as expected from the performance of SKM and AllocTC where higher slices can share all unused resources from the lower ones while this is not possible in RDM and MAM. Therefore, RDM and MAM had the lowest performance as they result in 89.13% for U and 66.07% for AR.

H. SCENARIO 4: PERFORMANCE IN ONLINE MODE UNDER NSF TOPOLOGY

In this online scenario we investigate SKM performance on limited resources of NSF network under different traffic loads and under fixed demands lifetime, in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov} compared to MAM, RDM and AllocTC. Moreover, in this scenario we used the same experiments that were considered in the second scenario.

1) SIMULATION SCENARIO SETTINGS

In order to evaluate our solution, the simulated scenario uses different traffic sources and different destinations on the NSF network consisting of 14 nodes and 21 links as shown in Fig. 12. The capacity of the link is equal to $R(l_{i,j}) = 150$ units. Moreover, the link resources are divided into four slices; each slice has $RCc(l_{i,j}) = 50$ units. As for the routing step, using the k-shortest path,the maximum value of k is set to 10. In all experiments of this simulation scenario, the demands are generated with a fixed lifetime equal to 1-time slot and the size of each demand is also fixed equal to 1 unit as the minimum granularity for allocation. Each demand has single priority generated in a random manner from (1 to 3) with a generation rate of demands per each unit time equal to 4000 demand. The total number of demands among slice generated until 10 unit time is 40,000 for each experiment.

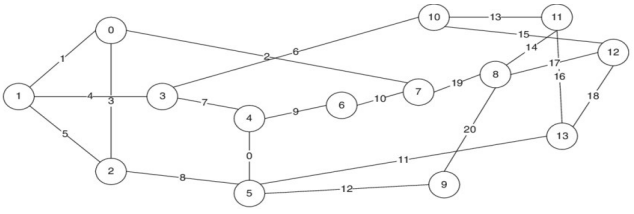


FIGURE 12. NSF topology.

2) RESULTS EVALUATION

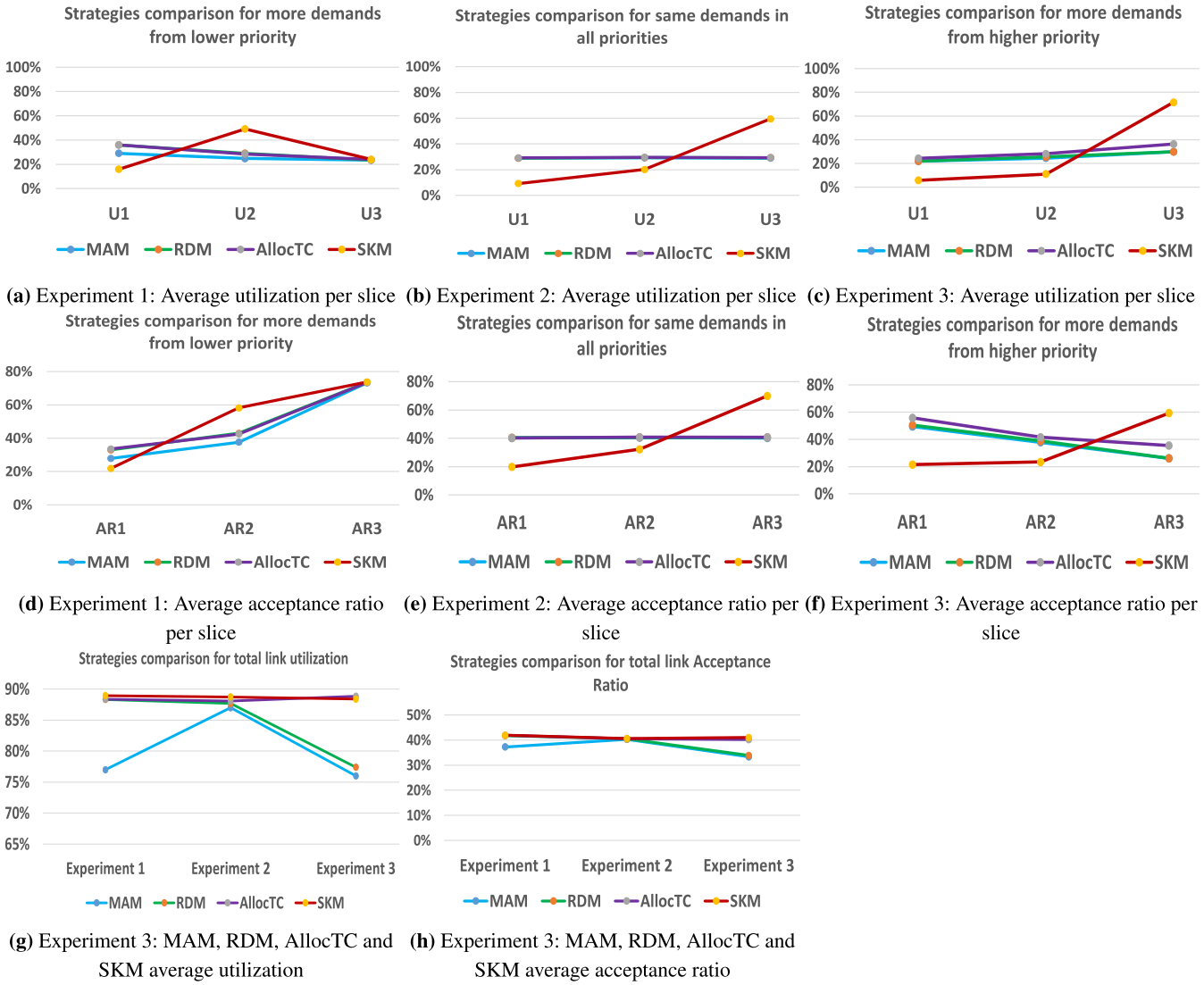
Table 9 shows the traffic load consideration (number of demanded resources in each slice) for the validation experiments in each slice in each unit time. Please note that, in all experiments, the capacity of each slice along the network is 1050 units ($RC_c(l_{i,j})$ * 21 links = total size of the slice across the network).

Figs. 13- 14 show the results of each algorithm in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov} using different traffic load according to experiments 1–3.

In terms of U and AR, Fig. 13g and Fig. 13h, illustrate the results from the three experiments for the MAM,

TABLE 9. Simulation experiments for the fourth scenario.

Number of slices in the generating file per-each unit time	Experiment 1 Load volume Traffic (Number of demands)	Experiment 2 Load volume Traffic (Number of demands)	Experiment 3 Load volume Traffic (Number of demands)
Slice-Type 1	2000	1500	500
Slice-Type 2	1333	1333	1334
Slice-Type 3	500	1500	2000



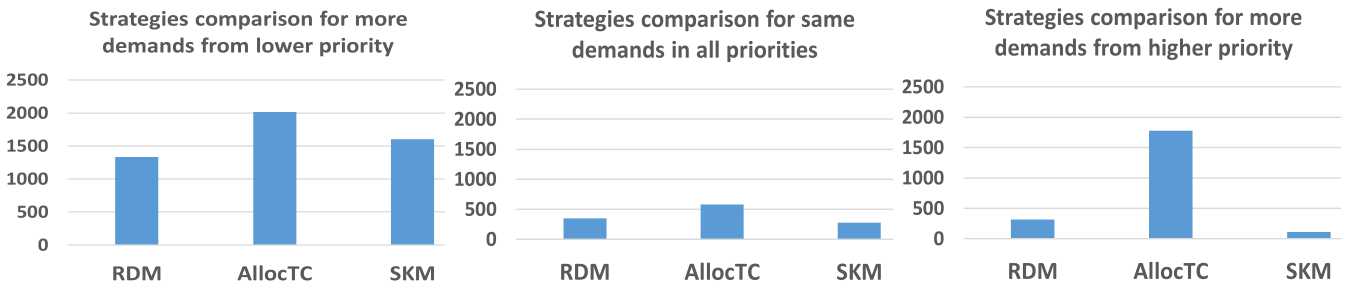
(a) Experiment 1: Average utilization per slice **(b)** Experiment 2: Average utilization per slice **(c)** Experiment 3: Average utilization per slice

(d) Experiment 1: Average acceptance ratio per slice **(e)** Experiment 2: Average acceptance ratio per slice **(f)** Experiment 3: Average acceptance ratio per slice

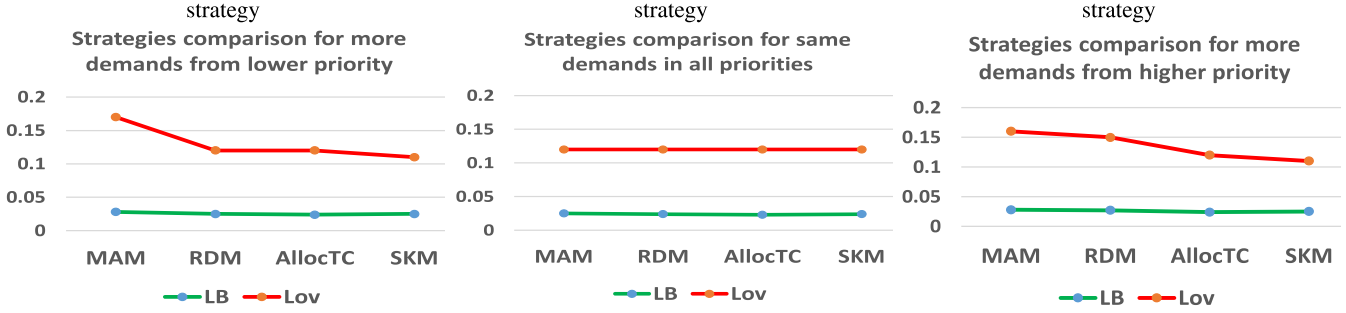
(g) Experiment 3: MAM, RDM, AllocTC and SKM average utilization **(h)** Experiment 3: MAM, RDM, AllocTC and SKM average acceptance ratio

FIGURE 13. Comparison of utilization and acceptance ratio in fourth scenario.

RDM, AllocTC and SKM. **In the first experiment**, SKM, AllocTC and RDM result in 88.93% U and 41.97% AR where 1679 demands are accepted from 4000 demands per each unit time. On the other hand, MAM achieves the lowest performance and results in 77.24% U and 37.30% AR where 1492 demands are accepted from 4000 demands per each unit time. **In the second experiment**, SKM, AllocTC, RDM and MAM result in 88.72%, 88.07%, 87.66%, 87% of U and 40.62%, 40.54%, 40.52%, 40% of AR, respectively since the load was the same in all slices. **In the third experiment**, SKM and AllocTC have similar performance in terms of U and AR as they both achieve 88.65% U and 41.05% where 1642 demands are accepted from 4000 demands per each unit time. On the other hand, RDM and MAM performance is the lowest one among the four strategies by achieving 77.36%, 76% in terms of U and 33.90%, 33% respectively in terms of AR. This is because there is no ability to share resources among the slices.



(a) Experiment 1: Preempted demands by each strategy **(b)** Experiment 2: Preempted demands by each strategy **(c)** Experiment 3: Preempted demands by each strategy



(d) Load balancing and overloaded link by each strategy **(e)** Load balancing and overloaded link by each strategy **(f)** Load balancing and overloaded link by each strategy

FIGURE 14. **Comparison of preempted demands, load balancing and overloaded link in fourth scenario.**

a: CONSIDERING HIGH LOAD IN LOWER PRIORITY SLICES

SKM, AllocTC, RDM and MAM have similar behavior in terms of both U3 and AR3 as they both achieve 23.98% and 73.72%, respectively because of the load distributions on slice 3 across the network being lower than its capacity. Moreover, SKM outperforms AllocTC, RDM and MAM by 20.14%, 20.61% and 24.32% in terms of U2. Further, in terms of ARc, SKM, achieves 12.5% for slice 2 more than MAM, RDM and AllocTC which achieves 15.29%, 15.71%, 20.63%, respectively due to the kicking operation (see Fig. 13a and Fig. 13d). Furthermore, Fig. 14a reveals that SKM, AllocTC and RDM resulted in 1601, 2012 and 1331 respectively in terms of P_{re} due to the kicking and preemption operations as we explained earlier. Moreover, Figs. 14d illustrate that SKM, AllocTC and RDM have a very close performance in terms of LB and L_{ov} as they result in 0.025 and 0.12, respectively due to the algorithms having similar utilization performance. On the other hand, MAM gives the lowest performance in terms of LB and L_{ov} as it results in 0.028 and 0.17, respectively, where more links are not fully used across the network.

b: CONSIDERING SAME LOAD IN ALL SLICES

Fig. 13 shows that the SKM outperforms MAM, RDM and AllocTC in the highest priority slice by 30.14% in terms of U3 and 29.26% in terms of AR3 due to the kicking operation as shown in Fig. 13b and Fig. 13e. Moreover, Fig. 16b shows that the SKM outperforms AllocTC and RDM by 298 and 71 respectively in terms of P_{re} due to the kicking operation. Furthermore, Fig. 14e shows that the performance of SKM, AllocTC, RDM and MAM are similar in terms of LB and L_{ov} as they result in average 0.024 and 0.12, respectively due to the algorithms having similar utilization performance.

c: CONSIDERING HIGH LOAD IN HIGHER PRIORITY SLICES

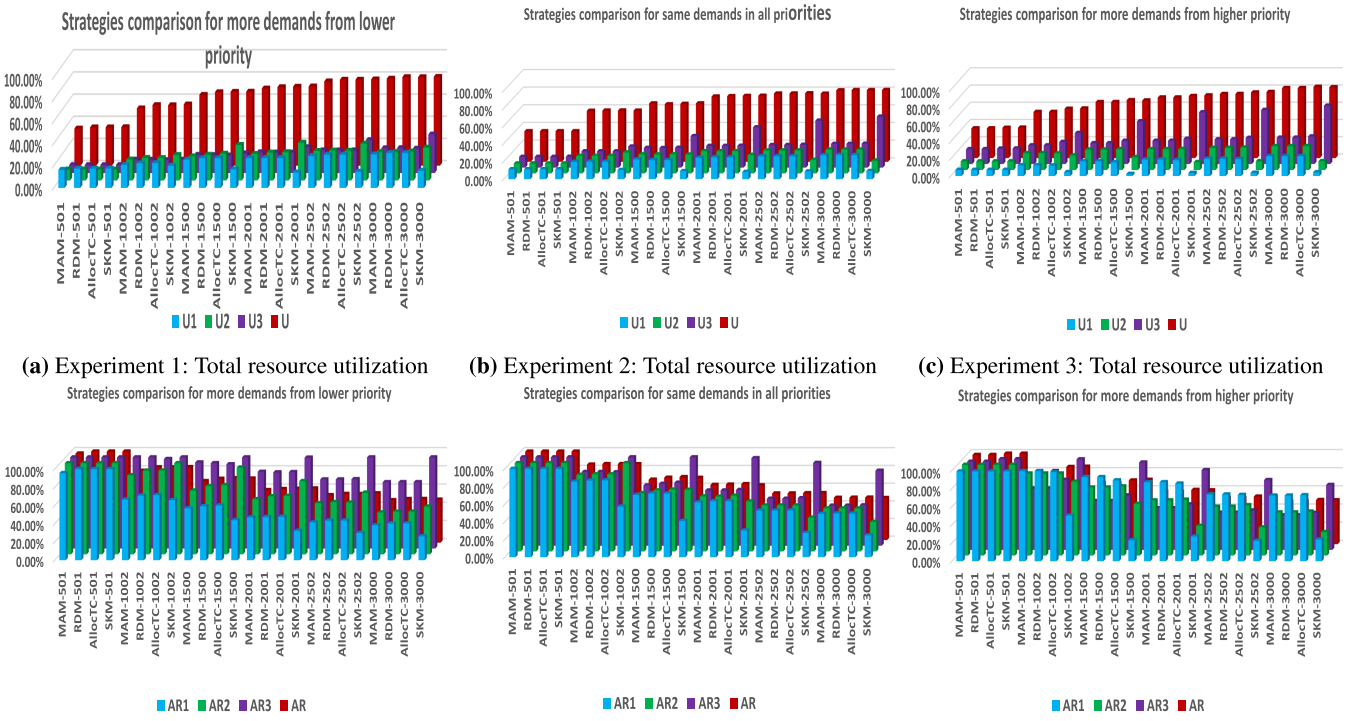
Fig. 13c and Fig. 13f illustrate that the SKM outperforms AllocTC, RDM and MAM in the highest priority slice by 35.17%, 41.71% and 41.71% in terms of U3 and by 23.8%, 33.19% and 33.19% respectively in terms of AR3 (as the expected from the behaviors) due to the kicking operation. Moreover, from the results of Fig. 14c, SKM outperforms RDM and AllocTC by 1665 and 205 respectively in terms of P_{re} since the load is too low on the lower priority slices, so, there is no need to use the kicking operation of SKM. Furthermore, Fig. 14f illustrate that SKM and AllocTC have a similarly good performance as they achieve 0.025, 0.12 in terms of both LB and L_{ov}. Moreover, RDM and MAM give the worst performance in terms of LB and L_{ov} and result in 0.028 and 0.16, respectively, where more links are not fully used across the network.

I. SCENARIO 5: PERFORMANCE IN OFFLINE MODE UNDER NSF TOPOLOGY

In the case of the offline scenario, we used the same NFS network topology and settings for the fourth scenario but with infinite demand lifetimes and considering a number of demands that varies from 501 to 3000 for each experiment in the studied scenario (see Table 10).

TABLE 10. Simulation experiments for the fifth scenario.

Number of slices in the generating file per-each unit time	Experiment 1 Load volume Traffic (Number of varied demands)	Experiment 2 Load volume Traffic (Number of varied demands)	Experiment 3 Load volume Traffic (Number of varied demands)
slice-Type 1	250, 501, 750, 1001, 1251, 1500	167, 334, 500, 667, 834, 1000	100, 167, 250, 333, 417, 500
slice-Type 2	167, 334, 500, 667, 834, 1000	167, 334, 500, 667, 834, 1000	167, 334, 500, 667, 834, 1000
slice-Type 3	100, 167, 250, 333, 417, 500	167, 334, 500, 667, 834, 1000	250, 501, 750, 1001, 1251, 1500



(a) Experiment 1: Total resource utilization **(b)** Experiment 2: Total resource utilization **(c)** Experiment 3: Total resource utilization

(d) Experiment 1: Acceptance ratio **(e)** Experiment 2: Acceptance ratio **(f)** Experiment 3: Acceptance ratio

FIGURE 15. Comparison of utilization and acceptance ratio in fifth scenario.

1) RESULTS EVALUATION

Figs. 15- 16 show the results of each algorithm in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov} when using different traffic load according to experiments 1–3.

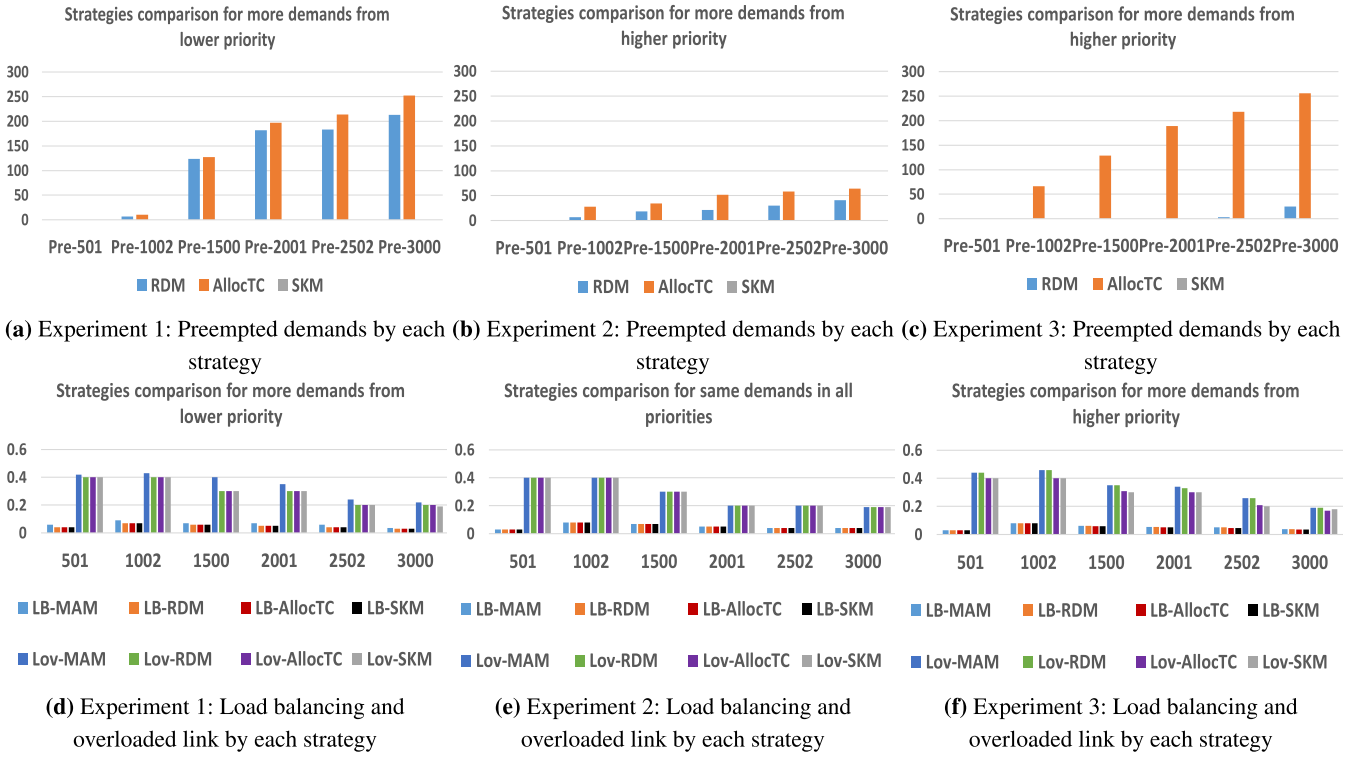
a: CONSIDERING HIGH LOAD IN LOWER PRIORITY SLICES

Fig. 15a and Fig. 15d reveal that as demand size increases as in the case when the size is 3000, SKM outperforms the other algorithms in terms of both U3 and AR3 by achieving 12.38% and 27.2% respectively. Moreover, SKM outperforms AllocTC, RDM and MAM by 3.46% in terms of U2 and by 5.5% in terms of AR2 due to the sorting operation of SKM. Furthermore, in terms of U and AR, SKM, AllocTC and RDM have similar behavior and result in 80.16% and 47.90% respectively. On the other hand, MAM gives the lowest performance because there is no sharing of unused resources between different priority slices which results in achieving 78.51% of U and 46.60% of AR.

Figs. 16a- 16c reveal that as demand size increases, irrespective of traffic load distribution for slices, SKM results in superior performance in terms of P_{re} compared to other algorithms as they achieve zero preempted demands due to the sorting operation of SKM. Moreover, Fig. 16d reveals that as demand size increases, as in the case where the size is 3000, SKM, AllocTC and RDM have similar performance in terms of LB and L_{ov} and result in 0.03 and 0.2, respectively since all links are fully used across the network. On the other hand, MAM gives the lowest performance in terms of LB and L_{ov} as it results in 0.036 and 0.22, respectively where more links are not fully used across the network.

b: CONSIDERING THE SAME LOAD IN ALL SLICES

Fig. 15b and Fig. 15e reveal that as demand size increases, as in the case where the size is 3000, SKM outperforms the other models in terms of both U3 and AR3 by 30.61% and 38.9%, respectively due to the sorting operation of SKM. Moreover, in terms of U and AR, all algorithms have similar



(a) Experiment 1: Preempted demands by each strategy

(b) Experiment 2: Preempted demands by each strategy

(c) Experiment 3: Preempted demands by each strategy

(d) Experiment 1: Load balancing and overloaded link by each strategy

(e) Experiment 2: Load balancing and overloaded link by each strategy

(f) Experiment 3: Load balancing and overloaded link by each strategy

FIGURE 16. **Comparison of preempted demands, load balancing and overloaded link in fifth scenario.**

performance as they result in 81.21% and 48.13%, respectively, because the number of demands was higher than all capacities of slices. Furthermore, Fig. 11d reveals that as demand size increases, as in the case where the size is 3000, all algorithms have similar performance in terms of both LB and L_{ov} as they result in 0.04 and 0.19 where more links are not fully used across the network.

c: CONSIDERING HIGH LOAD IN HIGHER PRIORITY SLICES

Fig. 15c and Fig. 5f reveal that as demand size increases, as as in the case where the size is 3000, SKM outperforms AllocTC, RDM and MAM in terms of AR3 by 35.68%, 37.14% and 37.14% and by 30.8%, 33.4% and 33.4% respectively in terms of AR3 due to the sorting operation of SKM. Moreover, in terms of U and AR, SKM and AllocTC have similar behavior as they both resulted in 82.63% and 48.70% where higher priority slices have greater demands for resources than other slices. Moreover, Fig. 16f reveals that as demand size increases, as in the case where the size is 3000, SKM and AllocTC have similar performance in terms of LB and L_{ov} and result in 0.036 and 0.17, respectively. On the other hand, MAM and RDM give the lowest performance in terms of LB and L_{ov} as they result in 0.039 and 0.19, respectively, where more links are not fully used across the network.

J. SUMMARY OF THE FINDINGS FROM THE SIMULATIONS

As observed from the simulation results of the above scenarios especially 2, 3, 4 and 5, the following points highlight the main findings:

1) THE EFFECT OF SKM PERFORMANCE COMPARED TO BAMs ON THE NETWORK TOPOLOGY

The proposed algorithm achieves up to 100% in terms of U in bandwidth-constrained environments. Therefore, the algorithm significantly enhances the user experience and resource utilization. Moreover, irrespective of the load division between slices, such as in scenario two, the algorithm resulted in 100% admission for the higher priority users whenever the resource requirements of the higher priority request does not exceed the available network resources as compared to a range of 38.54% for other algorithms (see Fig. 8b and Fig. 8e). To this effect, the proposed algorithm is well suited for emerging technologies such as network slicing that are constrained by strict QoS requirements and prioritized admission. Such technologies require dynamic allocation of resources and prioritized admission control.

2) THE IMPACT OF THE NETWORK TOPOLOGY ON THE PERFORMANCE OF THE ALGORITHMS

Despite NSF topology having more nodes and links, mesh topology provides better performance in terms of link utilization. This is attributed to the fact that all nodes are reachable within a single hop, and has a low betweenness centrality value compared to NSF, leading to fewer bottlenecks experienced by mesh topology. Since most demands are mapped on a single link path, minimal bottlenecks are experienced. Again, mesh topology provides for a high degree of connectivity due to the closeness of nodes and as a result, fewer links are used for mapping of each demand from source to

destination, thus, achieving improved AR, U, load balancing, resource consumption and the number of preempted demands performances across all the algorithms (see Fig. 8b, Fig. 8b3, Fig. 13b and Fig. 13e).

3) COMPARISON OF ONLINE AND OFFLINE SCENARIOS

The results of online scenario are better than that of the offline in terms of the total AR, since there is a chance for initially accepting low priority users which is not the case with offline scenario whenever the demanded resources of the high priority users exceed the available resources. In terms of average resources utilization, the offline scheme is higher than the online scheme. This is because the resources are unused in the initial stages (unit-times) for the online case. Considering 10 trails and obtaining the average value, the results obtained were as follows: In experiment 3 (more load in higher priority demands) of scenario 4 (NSF), SKM gives in online mode 88.84% of U, 5.75%, 11.07%, 71.57% of Uc, 41.05% of AR and 21.45%, 23.41%, 59.22% of ARc (see Fig. 13g, Fig. 13h, Fig. 13b and Fig. 13e). But in case of offline, with the increases of demand size (4000 demands) SKM gives 94.30% of U, 2.54%, 6.78%, 84.98% of Uc, 23.12% of AR and 3.63%, 8.23%, 14.16% of ARc.

4) TIME FOR EXECUTION

The proposed algorithm contains a sorting step, which introduces an extra overhead in terms of run time on the SKM algorithm. As an example, from the considered number of requests of experiment 3 in scenario 4, the average execution time in milliseconds for each admitted request is 53.87, 48.5, 36.23, 43.2, and 43.2 for the proposed algorithm and other algorithms that used AllocTC, RDM and MAM strategies respectively, averaged across all requests numbers. This result demonstrates that the proposed algorithm can process each demand in feasible time. Furthermore, the execution time for all algorithms increases with an increase in the number of requests. This is due to the additional complexity (e.g., the need for preemption / kicking actions) associated with the computation of additional paths to satisfy the different demands.

5) THE PROPOSED DEPLOYMENT ALGORITHM DRAWBACK

The algorithm needs to consider the aforementioned thresholds to define and guarantee the minimum resources for each slice which would avoid resources beat down for lower priority slices due to kicking process under congested scenarios.

VII. CONCLUSION AND FUTURE WORK

The paper discussed the problem of allocating resources to demands of different priority slices in a multi-slice network for both offline and online modes. This was based on the proposed SKM strategy for the purpose of effectively allocating available resources to serve demands in physical network paths. Since the computational load to find paths from the source node to destination node and their selection for the demands is huge even when using our proposed algorithm, then, the algorithm is forwarded to NFV architecture in order to provide the huge computational capacity required for the network service. SKM can be adapted to allocate bandwidth resources and any general resource management where resources require a reservation in addition to allocation stages, between different entities such as NFV service chain allocation and, of course, network slicing in future networks. Simulation results showed that thanks to our proposed algorithm, not only can we significantly improve the overall network usage, but also achieve the appropriate QoS and prioritized admission control for different E2E slice users. Moreover, our proposed algorithm can accept demands of considerable size, hence, guaranteeing a high admission of higher priority slices compared to other efficient schemes. This is mainly because the proposed algorithm implements a policy for resources selection that tends to increase the resources usage efficiency. Besides, it was proven that the algorithm is scalable with increasing substrate network demand sizes.

As a future addition, SKM will be improved by looking at the above-mentioned thresholds to identify and ensure minimum resources for each slice which would prevent resources beat down for lower priority slices. Moreover, we intend to perform a heuristic to provide a speedy response, which is critical in 5G networks.

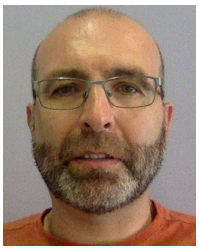
REFERENCES

[1] *System Architecture for the 5G System; (Release 16)*, document TS 23.501 V16.4.0, 3GPP, 2020.
[2] *Service Requirements for the 5G System; (Release 17)*, document TS 22.261 V17.2.0, 3GPP, 2020.
[3] L. Feng, Y. Zi, W. Li, F. Zhou, P. Yu, and M. Kadoch, "Dynamic resource allocation with RAN slicing and scheduling for uRLLC and eMBB hybrid services," *IEEE Access*, vol. 8, pp. 34538–34551, 2020, doi: 10.1109/ACCESS.2020.2974812.
[4] A. Huang, Y. Li, Y. Xiao, X. Ge, S. Sun, and H.-C. Chao, "Distributed resource allocation for network slicing of bandwidth and computational resource," in *Proc. ICC - IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149296.
[5] Y. Ma, W. Liang, J. Wu, and Z. Xu, "Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 393–407, Feb. 2020, doi: 10.1109/TPDS.2019.2937524.
[6] Y. Ma, W. Liang, J. Li, X. Jia, and S. Guo, "Mobility-aware and delay-sensitive service provisioning in mobile edge-cloud networks," *IEEE Trans. Mobile Comput.*, early access, Jul. 2, 2020, doi: 10.1109/TMC.2020.3006507.
[7] N. Reyhanian, H. Farmanbar, S. Mohajer, and Z.-Q. Luo, "Joint resource allocation and routing for service function chaining with in-subnetwork processing," in *Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 4990–4994, doi: 10.1109/ICASSP40776.2020.9054706.
[8] A. M. Medhat, G. A. Carella, M. Pauls, and T. Magedanz, "Orchestrating scalable service function chains in a NFV environment," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Bologna, Spain, Jul. 2017, pp. 1–5, doi: 10.1109/NETSOFT.2017.8004207.
[9] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5G: Challenges and opportunities," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 20–27, 2017, doi: 10.1109/MIC.2017.3481355.
[10] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: Enabling enterprises' own software-defined cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 146–153, Jul. 2016, doi: 10.1109/mcom.2016.7509393.
[11] "Network functions virtualisation (NFV); architectural framework," ETSI GSNFV, Tech. Rep., 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

[12] *OSM Release FIVE Technical Overview*, ETSI, Sophia Antipolis, France, 2019.
[13] M. Leconte, G. S. Paschos, P. Mertikopoulos, and U. C. Kozat, "A resource allocation framework for network slicing," in *Proc. IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 2177–2185, doi: 10.1109/INFOCOM.2018.8486303.
[14] *Description of Network Slicing Concept*, NGMN Alliance, San Diego, CA, USA, Jan. 2016. [Online]. Available: https://www.ngmn.org/uploads/media/160113_Network_Slicing_v1_0.pdf
[15] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017, doi: 10.1109/MCOM.2017.1600940.
[16] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management and prioritization in 5G mobile systems," in *Proc. 22th Eur. Wireless Conf.*, Oulu, Finland, 2016, pp. 1–6.
[17] S. Xiao and W. Chen, "Dynamic allocation of 5G transport network slice bandwidth based on LSTM traffic prediction," in *Proc. IEEE 9th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Nov. 2018, pp. 735–739, doi: 10.1109/ICSESS.2018.8663796.
[18] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Resource sharing efficiency in network slicing," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 909–923, Sep. 2019, doi: 10.1109/TNSM.2019.2923265.
[19] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "How should i slice my network: A multi-service empirical evaluation of resource sharing efficiency," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 191–206, doi: 10.1145/3241539.3241567.
[20] C. Song, "Machine learning enabling traffic-aware dynamic slicing for 5g optical transport networks," in *Proc. Conf. Lasers Electro-Optics (CLEO)*, San Jose, CA, USA, Aug. 2018, pp. 1–2. [Online]. Available: https://ieeexplore.ieee.org/document/8427129
[21] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *Proc. IEEE Conf. Comput. Commun.*, Atlanta, GA, USA. May 2017, pp. 1–9, doi: 10.1109/INFOCOM.2017.8057230.
[22] B. Han, J. Lianghai, and H. D. Schotten, "Slice as an evolutionary service: Genetic optimization for inter-slice resource management in 5G networks," *IEEE Access*, vol. 6, pp. 33137–33147, 2018, doi: 10.1109/ACCESS.2018.2846543.
[23] A. El-mekkawi, X. Hesselbach, and J. R. Piney, "Squatting and kicking model evaluation for prioritized sliced resource management," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 107006, doi: 10.1016/j.comnet.2019.107006.
[24] A. Bahnasse, F. E. Louhab, H. Ait Oulahyane, M. Talea, and A. Bakali, "Novel SDN architecture for smart MPLS traffic engineering-DiffServ aware management," *Future Gener. Comput. Syst.*, vol. 87, pp. 115–126, Oct. 2018, doi: 10.1016/j.future.2018.04.066.
[25] R. F. Reale, R. M. D. S. Bezerra, and J. S. B. Martins, "Applying autonomy with bandwidth allocation models," *Int. J. Commun. Syst.*, vol. 29, no. 13, pp. 2028–2040, Sep. 2016, doi: 10.1002/dac.3157.
[26] T. Taleb, B. Mada, M.-I. Corici, A. Nakao, and H. Flinck, "PERMIT: Network slicing for personalized 5G mobile telecommunications," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 88–93, May 2017, doi: 10.1109/MCOM.2017.1600947.
[27] T. Li, X. Zhu, and X. Liu, "An End-to-End network slicing algorithm based on deep Q-Learning for 5G network," *IEEE Access*, vol. 8, pp. 122229–122240, 2020, doi: 10.1109/ACCESS.2020.3006502.
[28] J. Wu, M. Wang, and C. Yuen, "Energy-aware concurrent multipath transfer for real-time video streaming to multihomed terminals," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6, doi: 10.1109/ICC.2016.7511547.
[29] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Energy-minimized multipath video transport to mobile devices in heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1160–1178, May 2016, doi: 10.1109/JSAC.2016.2551483.
[30] P. T. A. Quang, K. D. Singh, A. Bradai, and A. Benslimane, "QAAV: Quality of service-aware adaptive allocation of virtual network functions in wireless network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422757.
[31] V. Eramo and F. G. Lavacca, "Processing and bandwidth resource allocation in multi-provider NFV cloud infrastructures interconnected by elastic optical networks," in *Proc. 20th Int. Conf. Transparent Opt. Netw. (ICTON)*, Bucharest, Romania, Jul. 2018, pp. 1–6, doi: 10.1109/ICTON.2018.8473708.
[32] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1562–1576, Aug. 2018, doi: 10.1109/TNET.2018.2842798.
[33] F. F. Le and W. Lai, *Maximum Allocation Bandwidth Constraints Model for Diffserv-Aware MPLS Traffic Engineering*, document RFC 4125, Jun. 2005. [Online]. Available: https://www.hjp.at/doc/rfc/rfc4125.html
[34] F. Le Faucheur, *Russian Dolls Bandwidth Constraints Model for Diffserv-Aware MPLS Traffic Engineering*, document RFC 4127, Jun. 2005. [Online]. Available: https://www.hjp.at/doc/rfc/rfc4127.html
[35] R. F. Reale, W. D. C. P. Neto, and J. S. B. Martins, "AllocTC-sharing: A new bandwidth allocation model for DS-TE networks," in *Proc. 7th Latin Amer. Netw. Oper. Manage. Symp.*, Quito, Ecuador, Oct. 2011, pp. 1–4, doi: 10.1109/LANOMS.2011.6102265.
[36] S. K. Sadon, N. M. Din, M. H. Al-Mansoori, N. A. Radzi, I. S. Mustafa, M. Yaacob, and M. S. A. Majid, "Dynamic hierarchical bandwidth allocation using russian doll model in EPON," *Comput. Electr. Eng.*, vol. 38, no. 6, pp. 1480–1489, Nov. 2012, doi: 10.1016/j.compeleceng.2012.05.002.
[37] J. Socrates-Dantas, R. M. Silveira, D. Careglio, J. R. Amazonas, J. Sole-Pareta, and W. V. Ruggiero, "Novel differentiated service methodology based on constrained allocation of resources for transparent WDM backbone networks," in *Proc. Brazilian Symp. Comput. Netw. Distrib. Syst.*, Florianopolis, Brazil, May 2014, pp. 420–427, doi: 10.1109/SBRC.2014.50.
[38] N. Subhashini, "User prioritized constraint free dynamic bandwidth allocation algorithm for EPON networks," *Indian J. Sci. Technol.*, vol. 8, no. 1, pp. 1–7, Jan. 2015.
[39] R. Trivisonno, R. Guerzoni, I. Vaishnavi, and A. Frimpong, "Network resource management and QoS in SDN-enabled 5G systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–7, doi: 10.1109/GLOCOM.2015.7417376.
[40] C. Tata and M. Kadoch, "CAM: Courteous bandwidth constraints allocation model," in *Proc. ICT*, Casablanca, Morocco, May 2013, pp. 1–5, doi: 10.1109/ICTEL.2013.6632149.
[41] R. F. Reale and M. S. Romildo, "A bandwidth allocation model provisioning framework with autonomic characteristics," *Int. J. Comput. Netw. Commun.*, vol. 5, no. 6, p. 103, 2013.
[42] Reale, "A preliminary evaluation of bandwidth allocation model dynamic switching," *Int. J. Comput. Netw. Commun.*, vol. 6, no. 3, p. 131–143, 2014.
[43] A. El-mekkawi, X. Hesselbach, and J. R. Piney, "Network function virtualization aware offline embedding problem using squatting-kicking strategy for elastic optical networks," *Proc. 20th Int. Conf. Transparent Opt. Netw. (ICTON)*, Bucharest, Romania, 2018, pp. 1–10, doi: 10.1109/ICTON.2018.8473869.
[44] A. El-mekkawi, X. Hesselbach, and J. R. Piney, "A novel admission control scheme for network slicing based on squatting and kicking strategies," in *Proc. 12th Int. Conf. Transparent Opt. Netw. (JITEL)*, Zaragoza, Spain, 2019, p. 1–8.
[45] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012, doi: 10.1109/TNET.2011.2159308.
[46] G. Kibalya, J. Serrat, J.-L. Gorricho, H. Yao, and P. Zhang, "A novel dynamic programming inspired algorithm for embedding of virtual networks in future networks," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107349, doi: 10.1016/j.comnet.2020.107349.

AHMED EL-MEKKAWI received the M.S. degree in electronics engineering and communications from the Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt, in 2016. He is currently pursuing the Ph.D. degree with the Network Engineering Department, Universitat Politecnica de Catalunya (UPC), Barcelona, Spain. He previously worked as the Director of the Information Network with the Department of Passports and Immigration, Egyptian Interior Ministry, from 2007 to 2016. His current research interests include networks virtualization, resources management, broadband networks, quality of service, software-defined networks, and bandwidth allocation.mechanisms.

XAVIER HESSELBACH (Senior Member, IEEE) received the M.S. and Ph.D. degrees (Hons.) in telecommunications engineering from the Universitat Politecnica de Catalunya (UPC), in 1994 and 1999, respectively. In 1993, he joined the Design, Modelling and Evaluation of Broadband Networks Group with the Network Engineering Department, UPC. He is currently working as an Associate Professor with the Department of Network Engineering (Department Enginyeria Telematica), UPC. He has been involved in several national and international projects. He has taken part in several European and Spanish research projects, such as the EuroNGI/FGI/NF Network of Excellence, COST293, Mantychore, and All4Green, being main UPC researcher in the Mantychore and All4Green projects. He is the author of four books and more than 100 national and international publications in conferences and journals. His research interests include networks virtualization, resources management, broadband networks, quality of service, and green networking. He has participated in the technical program committees of several conferences. In 1994, he received the award for the Best Master Thesis on Networks and Telecommunication Services from the COIT/AEIT, Spain. He was the Information Systems and Internet Chair in Infocom 2006, and a Guest Editor of the *Ad Hoc Networks* Journal and the *Journal of Electrical and Computer Engineering*.



JOSE RAMON PINEY received the B.S. degree in electronic engineering and communications from the Universidad de las Américas, Puebla, México, in 1989, the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, in 1993, and the Ph.D. degree in telematics engineering from the Technical University of Catalonia, Barcelona, Spain, in 2006. Since 1999, he has been an Associate Professor of Telematics Engineering, Technical University of Catalonia. His research interests include passive optical networks, software defined networks, and bandwidth allocation mechanisms.