UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
E scola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

PhD Thesis

Resource Allocation and Management under Priorities Based on the Squatting-Kicking Model for Multi-Slice 5G Networks

A dissertation submitted in partial fulfillment of the requirements for the degree Doctor of Philosophy in Network Engineering

Supervisor: Dr. Xavier Hesselbach

Co-supervisor: Dr. Jose Ramon Piney

Author: Ahmed El-mekkawi

Dept. Network Engineering

Universitat Politecnica de Catalunya (UPC)

Barcelona, Spain, 2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Resource Allocation and Management under Priorities Based on the Squatting-Kicking Model for Multi-Slice 5G Networks

ISSN not assigned yet
This thesis was written in LaTeX using the class PhDThesisPSnPDF (v2.3.1).
It contains a total of 177 pages. Compiled with pdfLaTex on Wednesday 2^{th} June, 2021.

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Acknowledgment

I am humbly proud to acknowledge the extensive support of my advisor, Professor Xavier Hesselbach and my Co-advisor, Professor Jose Ramon Piney. This thesis would not have been complete without his in-depth contributions, invaluable mentoring and support, numerous brainstorming sessions, and continuous advice throughout my PhD training. Were it not for vigorous follow-up and encouragement, this thesis might not have ended.

Moreover, I am very grateful for the extensive help and guidance from the administration office of the Network Engineering department at UPC.

Finally, I would like to acknowledge the great help, support and inspiration I have received from my PhD colleagues in the Network Engineering department at UPC. They were my extended family outside.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Dedication

I express my sincere gratitude to my mom, Doria for her constant and unwavering love, support and understanding while pursuing my PhD which made thesis completion possible. She was always there sometimes, I thought impossible to keep going, and it helped me keep focus and work hard. I greatly appreciate her contributions and her belief in me. I appreciate my sisters Ghada and Noha for enduring my ignorance and patience that they showed during my absence, and I appreciate my family for their calling and support.

Words will never say how grateful I am to all of you. I consider myself the most fortunate in the world to have such a loving and caring family, standing by my side with their unconditional love and support.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Executive Summary

The aim of the upcoming Beyond Fifth Generation (B5G) networks is to meet network services characterized by low latency and high reliability among others in different slices in order to provide a high-quality user experience [Clemm (2020), ITU-T (2019)]. However, existing best-effort networking schemes that implement traditional methods of controlling and allocating network and computing resources do not meet such strict service requirements well. In International Telecommunication Union-Telecommunication sector (ITU-T) [A. Karimi (2020)], future services are defined as Network 2030 Services under a chartered Focus Group on Networks 2030 (FG-NET2030). The results from the FG-NET2030 analysis suggests that current networks cannot accommodate real-time applications with low latency and high bandwidth requirements. Moreover, current networks lack the capabilities to dynamically aggregate and share network resources through multiple flows, which is essential for future services.

However, in order to satisfy the strict requirements of those services, intelligent algorithms and techniques that incorporate 5G enablers are needed to introduce novel network management systems. These intelligent algorithms shall not only result in efficient utilization of network resources, but also guarantee the required quality of service for the priority slices. Moreover, cognizant of the strict latency requirements of the different services, such algorithms should include delay constraints of requests [A. Clemm (2020), C. Huang (2020)].

Despite the advantages expected from future services are real-time applications, that should benefit from reduced physical and logical paths between end users and data or service hosts [M. M. Hussain (2019), P. Bellavista (2017), C. Kuo (2017)]. However, all the above requirements are not intended for the network slicing paradigm alone. Therefore, in addition to network slicing, we want to leverage technologies and components that have features such as network programming, dynamic network reconfiguration and orchestration to enable improved performance and efficient resource management. Such technologies include Network Function Virtualization (NFV) and Software Defined Networking (SDN) among others.

Consequently, the main objective of this PhD thesis is to develop a service deployment algorithm that uses Squatting and Kicking techniques intelligence to effectively allocate, manage, and control slice resources under several constraints in a real-time multi-slice scenario, such as priority, bandwidth, and E2E delay with targeting to maximize the total resource usage in the substrate network.The proposed online algorithm, allocates the available resource to different priority demands from source

node to destination node along the routed path according to more realistic constraints, such as links' bandwidth and end-to-end delay. Moreover, the benefits of the new proposed algorithm will be reflected on creating real-time demands for 5G applications that are sensitive to delay, in addition to solving the resource allocation problem for large scale networks, using fewer resources and generating lower costs. Further, the proposed algorithm is adaptable to meet various QoS requirements of services, guaranteeing high QoS levels and providing high admission for higher priority classes under congested scenarios.

In terms of managing bandwidth resources in a multi-slice scenario, Bandwidth Allocation Models (BAMs) offer improved metrics over best effort models. The proposed algorithm outperforms the others by far especially, during congested scenarios. To this end, this thesis proposes a resource allocation model called Squatting and Kicking model (SKM) to maximize the number of successfully embedded demands while maximizing the utilization in the multi-slice networks by choosing less congested paths through the efficient allocation of demands on the network. In particular, the thesis mainly focused on the problem of dynamic and efficient allocation of link bandwidth resources to services with different priorities, thereby improving the service quality based on squatting and kicking techniques. SKM, solves network resource embedding problem for offline and online scenarios under distinct network profile/state, such as heavy traffic loads, dynamic traffic and application scenarios. Moreover, this thesis analyzes the impact of delay constraint on the performance of an online resource allocation algorithm based on an intelligent efficient SKM, proved in this work to be the most effective up to the present time yet.

The proposed algorithm incorporates kicking and squatting of resources as innovative techniques enabling it to achieve 100% resource utilization and acceptance ratio for higher priority slices in scenarios where the other state of art algorithms can not reach by far in some scenarios. Simulation results showed that incorporating delay constraints has a significant impact on the performance of all considered algorithms including the proposed algorithm, resulting in up to 10% and 4% reduction in terms of average resource utilization and acceptance ratios respectively.

Nevertheless, this thesis suggests that future enhancements for the proposed algorithm, need to be focused around modifying the proposed squatting and kicking techniques by considering thresholds to define and guarantee minimum resources for each slice that will avoid resources beat down for lower priority slices. This is because , SKM intends to favor users belonging to high priority slices in terms of admission and resource allocation hence the observed superior performance for high slices at the expense of low priority slices. Moreover, we intend to perform a heuristic to provide a speedy response, which is critical in 5G networks. Furthermore, future work could improve the proposed algorithm by incorporating machine learning techniques for smart traffic and optimal path prediction, and also executing machine learning for higher resource efficiency, faster load balancing, and more precise resource allocations based on a variety of quality of service metrics.

The structure of this thesis includes six chapters. **Chapter 1** covers the main objectives of the thesis, and it points to some observations about resource allocation efficiency for physical substrate network.

In addition, it highlights the main research questions, and concludes with summary of the main thesis contributions. Then **Chapter 2** provides detailed background review and coverage, for the related literature about resource allocation problem in large scale networks, and about 5G core network technologies, including software defined networks, network function virtualization, and network slicing.

Chapter 3 provides a detailed presentation about our proposed solution methodology for the resource allocation problem, clarifying its general modelling and explaining the proposed squatting and kicking techniques to solve the resource allocation problem to serve demands belonging to slices with different priority. In addition, this chapter provides step-by-step problem formulation and analysis of the proposed algorithm, compared to other algorithms, which will be the main building block of business in Chapters 4 and 5.

In **Chapter 4,** an intelligent service deployment algorithm that uses SKM strategy from chapter 3 is introduced to maximize the number of successfully allocated demands while maximizing the utilization and balancing the load considering a full multi-slice network by choosing less congested paths based on the computation algorithm executed in the NFV architecture, through the efficient allocation of demands on the network for online and offline scenarios. The chapter introduces the problem formulation, explains the proposed new online and offline algorithms, and concludes by showing the simulation results, and it includes evaluations about the proposed algorithm performance in terms of resource utilization, acceptance ratio, the total number of preempted demands, load balancing, overloaded link across different network topology complexities.

Moreover, in **Chapter 5**, a new service deployment algorithm is proposed that uses the intelligence of squatting and kicking techniques for an online scenario, and its main goal is to accept requests among different priority slices, in real-time, while maximizing the total resource utilization in the entire substrate network considering E2E delay as the primary allocation constraint. Additionally, the impact of E2E delay constraint on the performance of the proposed online deployment algorithm was deeply analyzed, representing direct application for network slices in future 5G networks and beyond. Then it concludes by showing the evaluation results for different network topology complexity scenarios.

In the last chapter, **Chapter 6,** a focused summary of the main findings and recommendation for suggested future research is introduced.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Contents

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

List of Figures

List of Tables

List of Abbreviations

3GPP	Third generation partnership project
5G	Fifth Generation
6G	Sixth Generation
AI	Artificial Intelligence
AP	Access Point
API	Application Programmable Interface
Alloc-TC	AllocTC-Sharing Model
B5G	Beyond Fifth Generation
BAM	Bandwidth Allocation Model
BFS	Breadth-First Search
BS	Base Station
CAC	Call Admission Control
CAPEX	Capital expenditure
CoS	Class of Service
DC	Data Center
DS-MPLS	DiffServ-aware Multi-Protocol-Label Switching
E2E	End-to-End
EON	Elastic Optical Network
EPON	Ethernet Passive Optical Network
ETSI	European Telecommunication Standard Institute
FG-NET2030	Focus Group on Network 2030
IETF	Internet Engineering Task Force
ION	Intelligent activity network
ITU-R	International Telecommunication Union Radio Communication Sector
ITU-T	International Telecommunication Union Telecommunications
IoT	Internet of things
KPI	Key performance indicator
LTE	Long Term Evolution
MAM	Maximum Allocation Model
MIoT	Massive Internet of Things
MNO	Mobile Network Operator
NCC	Network and computing convergence
NF	Network Function
NF	Network Function Virtualization Infrastructure
NFV	Network function virtualization

NFV-MANO	NFV Management and Orchestrating
NGMN	Next Generation Mobile Network
NS	Network Service
NV	Network virtualization
NetConf	Network Configuration Protocol
ONF	Open Networking Foundation
OPEX	Operational expenditure
OSPF	Open Shortest Path First
QoE	Quality of Experience
QoS	Quality of Service
RDM	Russian Doll Model
SC	Service Chain
SDN	Software Defined Networking
SDO	Standards-Development Organization
SKM	Squatting and Kicking Model
SLA	Service level agreement
SNMP	Simple Network Management Protocol
STIN	Space-terrestrial integrated network
TIRO	Internet for remote operations
TMF	TeleManagement Forum
URLLC	Ultra-Reliable Low-Latency Communication
V2X	Vehicle-to-everything
VNF	Virtual Network Function
VPN	virtual private network
eMBB	enhanced Mobile Broadband

Symbols

G(X, L)	Directed graph of the physical network
X	Substrate network nodes
L	Substrate network links
l	Single hop edge from substrate node i to j
$RC_c(l)$	Resource constraints for slice c also equal to maximum reservable resources for slice c in l
$(CT_c)(l)$	Priority slice c in l
R(l)	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
$R_a^t(l)$	Available resource capacity on l at time t
$R_z^t(l)$	The consumed resources capacity on l at time t
$X_w^{t,l}$	Is the binary variable equal to 1 if the demand $w \in W$ is assigned resources at link $l \in L$, 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
$\delta(l)$	Propagation delay
$\delta_{s,r}(l)$	Maximum acceptable E2E delay
D_c	Total number of demands by slice c
D	Total number of demands by all slices
D_{c_t}	Total number of demands arriving for each class for each unit time
$S_c(l)$	The actually allocated resources to slice c on l
c_{d_w}	Priority of demand w
BD	Number of blocked demands by all slices
BD_c	Number of blocked demands by slice c
$(BD)_t$	Number of blocked demands from the current unit time
$(BD_c)_t$	Number of blocked demands for class c from the current unit time
AD	Number of accepted demands by all slices
AD_c	Number of accepted demands by slice c

$(AD)_t$	Number of accepted demands from the current unit time
$(AD_c)_t$	Number of accepted demands for class c from the current unit time
μ	The mean value of the utilization of all links across the network
P_{LTH}	The number of preemption of higher priority traffic by lower priority traffic
P_{HTL}	The number of preemption of lower priority traffic by higher priority traffic
P_{re}	Number of preempted demands in the whole network
$SH_q(l)$	Squatted resources from higher priority $slice_q$ on l. Where $q \in [1, N]$
$SL_q(l)$	Squatted resources from lower priority $slice_q$ on l
$K_q(l)$	Kicked resources from lower priority $slice_q$ on l
$Z(d_w)$	1 if demand w is successfully mapped

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

CHAPTER 1

Introduction

Strong hopes to commercialize 5G by 2020 [J. Sachs (2018)] drove standardization agencies, academics, and businesses over the next decade after 5G. This has prompted all stakeholders to propose and forecast potential network services (Network 2030 Services) [FG-NET-2030 (2020), ITU-T FG-NET2030 (2019)] with explicitly articulated key performance indicators (KPI) and network QoS requirements. For example, ITU-FG-NET2030 T's defined a range of emerging applications with strict network service specifications and potential advanced technologies (intelligent network systems) that might not be well supported by current network infrastructure and technology [ITU-T FG-NET2030 (2019)].

Intelligent network systems, thus, reflect innovative technology and functionality well beyond those defined within the 5G network model by the Standards Development Organizations (SDOs) such as ESTI, 3GPPP and IETF among other [B. Chang (2019)], and thus lead to new advanced concepts and implementations of novel frameworks that are extremely versatile and willing to respond to the evolving requirements implemented by emerging potential applications. They need to support a modern set of continuously interconnected artifacts and devices and new forms of communicating with them. Computer and network resources can need to be handled independently to be effective, thus ensuring a high degree of coordination to satisfy demand requirements. Such a paradigm change indicates a network and infrastructure development past 5G, as shown in Fig. 1.1 below.

With these recognize vulnerabilities and obstacles, as well as the network and service development requirements imposed by potential applications, the research community is taking this as an incentive and inspiration for study. However, in order not to be overly optimistic, we have established a delay sensitive scenario among the network 2030 services according to 3GPP use cases. The 3GPP has identified enhanced Mobile Broadband (eMBB), massive Internet of things (MIoT), Ultra-Reliable and Low-Latency Communication (URLLC) and Vehicle-to-everything (V2X) as the four critical usage scenarios in 5G communication systems [3GPP1 (2020), 3GPP2 (2020)]. Among other factors, such a service needs extremely large processing capacity, exceptionally low latency, and tremendous transmission rates in order to maintain communication live and true [Clemm (2020), A. Clemm (2020), C. Huang (2020)]. The capacity of networks to serve underlying systems with a relatively low latency (quantifiable latency in-time services) and a sufficiently high degree of processing is not available in current networks [Clemm (2020), A. Karimi (2020)].

The network slicing paradigm has been suggested in order to address the above difficulties, with the

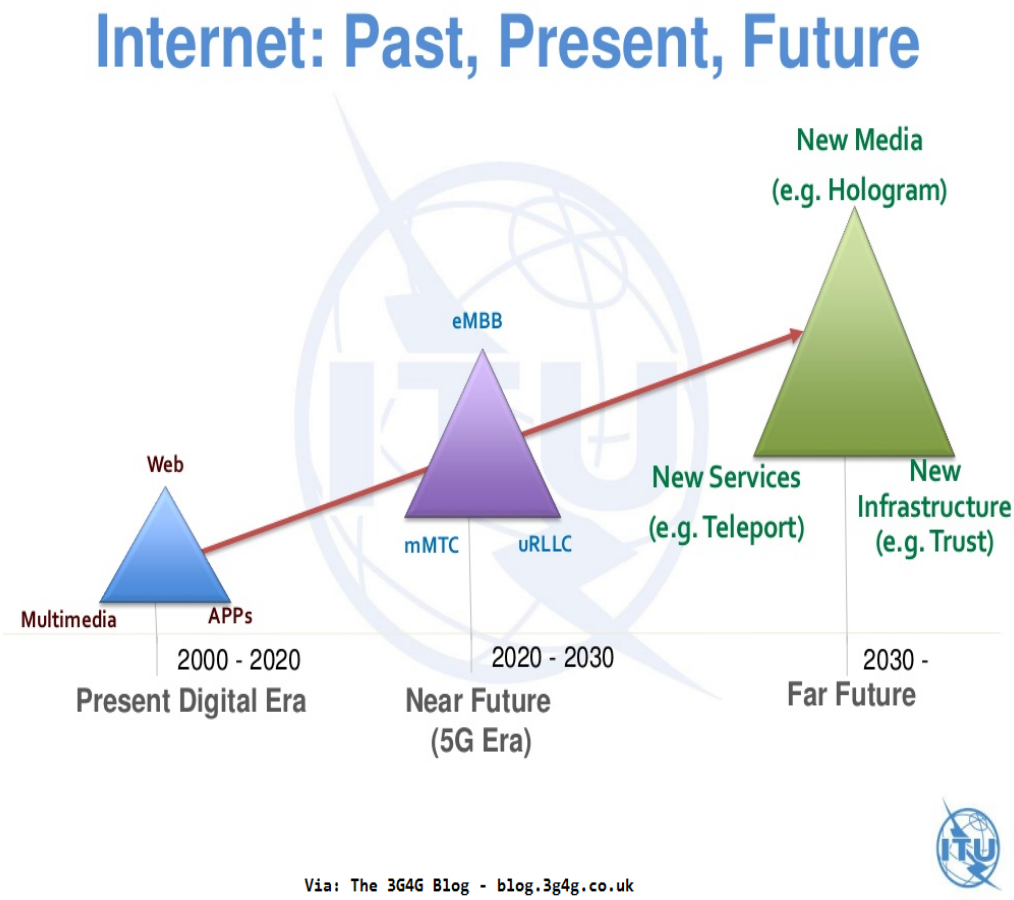


Figure 1.1: Network and network services development [ITU 'Network 2030' (2018)]

goal of taking infrastructure resources from the bandwidth, antennas and all backend networks and equipment and utilizing them to realize several sub-networks with various properties. In order to impose its own separate, no-compromise network for its preferred applications, each sub-network slices the resources from the physical network end-to-end (E2E).

However, the above scenario presents two major challenges in a number of dimensions: First, the network resources are limited and exhaustible which posses challenges regarding how to efficiently allocate these resources to the different slices while meeting the divergent service requirements such as delay and throughput; secondly, the different service slices are characterized by different priorities and criticality, which causes complexities regarding real-time E2E QoS routing of the different services while managing the prioritization levels across the different slices. This is more critical under network resource scarcity especially under disaster events and network congestion. In this context, the network is not able yet to take intelligent decisions in order to optimize the behaviour [Xu C (2019)].

Consequently, this scenario necessitate development of intelligent resource allocation algorithms with joint capability to: i) maximize the utilization of network resources while achieving possible maximum productivity and facilitation of sharing resources among slices while allowing a specific slice to meet the the demanded Service Level Agreement (SLA); ii) Guarantee prioritization of critical services especially under congested scenarios; and iii) Satisfy all the constraints related to the request especially end-to-end delay.

As a contribution to the above challenges, in this PhD thesis an efficient algorithm based on squatting and kicking techniques has been introduced. The squatting technique provides for 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. The results from the simulations revealed that the introduced algorithm was optimal in terms of resource allocation and QoS for high priority users and admission control while improving the total resource utilization.

In a realistic 5G scenario, the network topology is complex, the transmission is real-time, the requests arrive in online mode, and the services are delay constrained. 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. With this motivation, this PhD suggests an intelligent algorithm that uses the intelligence of SKM strategy for efficient deployment and allocation of network resources in a multi-slice scenario. We formally define the proposed algorithm to solve the problem of real-time resource allocation for QoS end-to-end routing considering realistic network behaviour by incorporating delay constraints and considering full network topology under online request arrival. In addition, we extensively analyse the impact of delay on the performance of the proposed algorithm. Moreover, although network slicing is envisaged to be implemented in an end-to-end fashion across links and nodes, the resources of the virtualized node functions can be scaled up with more ease compared to the link resources. Therefore, the link resources form the performance bottleneck of the network especially under bandwidth intensive applications. Cognizant of this fact, this thesis focuses on how intelligence can be deployed in NFV in order to provide efficient utilization of link bandwidth resources in a multi-slice scenario considering strong constraints as required in 5G networks. NFV technology accelerates the process of user-oriented services execution appearing in both cost and time saving by allowing execution and deployment of middleboxes as Virtual Network Functions (VNFs) running on Virtual Machines (VMs). Furthermore, NFV facilitates service deployment by employing the concept of service chaining (SC) [A. M. Medhat (2017)]. Specifically, the computational difficulty of finding the right paths from a source node to a destination node is enormous in 5G realistic scenarios, so the proposed algorithm is suggested as a suitable candidate to be implemented inside NFV in the form of SC that provides the enormous computational power needed by the network to make intelligent decisions about admission control, routing path computation and resource allocation.

In line with bandwidth resource management in a multi-slice scenario, BAMs provide enhanced metrics compared to best-effort models. SKM outperforms the others by far especially, during disaster

and congested scenarios.

In the following sections, a high level introduction about the overall thesis objectives will be presented, as well as the main motivation behind this thesis, followed by listing the main questions that this thesis will ultimately answer.

1.1 Thesis main objectives

The main objective of this PhD thesis is to:

Develop a novel intelligent algorithm to solve the resource allocation problem using advanced Bandwidth Models and Delay Constraints in multi-slice network

Specifically, the following tasks will be performed to realize the main objective:

1. Develop a QoS algorithm for multi-class networks based on squatting and kicking techniques that can be used with any networks such as Elastic Optical Network (EON), wireless network, MPLS and among others. SKM provides a new policy for selecting and serving demands, which takes QoS constraints into account for different priorities/classes.
2. Develop a paradigm based on NFV architecture to provide the massive computational capacity required in the network services and support the resource allocation strategy proposed for multiple slice networks based on resources utilization optimization using a proposed algorithm for offline and online cases.
3. A mathematical model that natively supports multiple network slices, which differ in terms of QoS requirements. The model was proposed taking 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, topology and scenarios. Besides, 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.
4. Evaluate the impact of including end-to-end delay as a main resource allocation constraint on the performance of the suggested online deployment algorithm were deeply analyzed, representing direct application for network slices in future 5G networks.

Moreover, promised performance enhancements by the proposed model (i.e., SKM) are listed as follows:

1. Optimized resource utilization through efficient allocation of the resource demands on the network.
2. Guarantees high admission of higher priority slices under different input traffic volumes (especially in congested scenarios). On the other hand, when the traffic is not congested the SKM behaves similar to BAMs.
3. Adaptability to emerging technologies that are characterized by diverse QoS requirements and prioritized admission control, especially under network slicing scenario.

1.2 Observations about resource efficiency for future physical networks

The fundamental building blocks of today's physical network and the potential Internet are made up of physical nodes and their links. The allocation of network services includes the allocation of links and node resources. Most of the works in the literature review are addressing core network resource allocation and there exist few works on link resource management with prioritized traffic classes. Moreover, management of link resources is the most important part of allocating network resources and presents new challenges in several dimensions (e.g., bandwidth allocation along links of a requested path, management of the prioritization on the links, and isolation between traffic classes) compared to node resources [S. Xiao (2018), C. Song (2018), C. Marquez (2018), C. Marquez (2019)]. Besides, the future network technologies are characterized by extremely wide bandwidth requirements that will be accessible by users under limited available bandwidth resources [C. Song (2018)]. Resource allocation problem along the routing path based on the bandwidth and priorities will need crucial and promising models to address the above challenges [S. Xiao (2018)]. Therefore, in this thesis we only focused on the problem of dynamic and efficient allocation of links and bandwidth resources to the services with different priorities and thereby improving the service quality. Moreover, we want to mention that our focus for the problem (i.e., bandwidth resource allocation) is different from and in fact more challenging than the problems in the literature in terms of congestion and path requests with bandwidth requirements along the demanded path. Our proposed algorithm acts stricter on priorities and significantly differentiates priorities under congested scenarios to improve the utilization and provide high admission for higher priority class users in terms of traffic, which is crucial for the quality of service guarantee.

Practical application scenarios:

The proposed algorithm is a suitable strategy for emerging technologies that are characterized by diverse QoS requirements and prioritized admission control. The concept of QoS allows certain types of traffic to be prioritized in the network. If some traffic, such as video, is more important than others in a network, then by using our algorithm, a network administrator can prioritize that video traffic to ensure that the service remains uninterrupted while the other traffic may be suspended or even dropped. Another example can be the emergency scenarios. Directly after an emergency incident, first responders (e.g., police, firefighters, medical personnel, among others) are sent to the incident area for rescue and relief operations. As the first minutes are vital to saving human lives, robust and ubiquitous communications should be available to first responders. Also, diverse QoS requirements are typical in the 5G network, which are expected to serve flexible and diversified requirements. Hence the need to allocate resources dynamically while respecting priorities is crucial. A case at hand will be network slicing scenario, where the different slices have varying priorities in terms of admission and resource allocation. Another application could be resource management in Virtual Network Embedding (VNE) scenario, where physical resources require sufficient reservation plus allocation phases to satisfy virtual demands on top of a substrate network that has limited residual capacities.

Therefore, the conducted literature review by this thesis highlighted the importance of exploring efficient ways to maximize total resource utilization of a full network while ensuring high levels of QoS requirements, considering the benefits of network virtualization, network slicing, as well as considering the following high-level observations:

1. The Virtual network embedding problem consists of solving the mapping between virtual resources and physical resources, while the dynamic resource allocation consists of opportunistically changing allocated resources on the basis of traffic demand. Network slicing has certain parallels with network virtualization and the same problem division can be considered.
2. It is possible to implement slicing with specific Quality of Service guarantees on 5G realistic networks.
3. The restriction we found is the embedding problem that we will be grappling with: i) The heterogeneity and the variability of the final customer and ii) The dynamic resource allocation often poses new challenges: not only the traffic load will change, but also the link capacity and the network load (number of connected devices). Therefore, the allocation of resources needs to be adjusted considering this new dynamism.
4. The problem of dynamic resource allocation is to design mechanisms to maintain or update the allocation in the event of changes, so as to ensure QoS and efficient use of resources.
5. The importance of defining a new method for multi-path routing that takes into account various QoS constraints such as priority, bandwidth and delay in order to meet the SLA requirements in real-world scenarios.
6. In the case of applying end-to-end delay as recommended for 5G systems under network slicing scenario, including end-to-end delay on the allocation process affect the extend of maximizing total resource utilization and acceptance ratio especially, during congested case.

1.3 Main research challenges

In light of the above observations, the proposed solution aims to smartly share resources among different slices of services according to their service needs and assigned priorities. Future networks (e.g., 5G) management faces a large number of challenges that can be addressed through our proposed model including but not limited to the following:

1. Admission control and resource allocation problems:
   With the fast growth of 5G networks demands and limitation of resources and also, 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 QoS requirements [3GPP1 (2020)]. Thus it motivates us 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.
2. The need for a preemption policy:
   The 5G networks should allow a flexible means to enforce prioritization among the services under congested case. The traffic prioritization may be enforced by adjusting resource utiliza-

tion or preempting lower priority traffic [3GPP2 (2020)]. In this regards, a new preemption policy (kicking technique) has been formulated that acts aggressively in congested scenarios for the proposed SKM technique [El-mekkawi (2020)]. In this work, we use kicking to imply the ability to remove resources from a lower priority class (slice), including both borrowed and those that are reserved for that class. Besides, traditional preemption act in BAMs differing from kicking by denotes expulsion of a lower priority class demand from resources it borrowed from other slices and not its reserved resources.

3. E2E QoS for a service:
   Setting up QoS policies in nodes along the path is a complicated task. However, another constraint must be handled; it is the routing with QoS constraints. In other words, how to ensure that the used path meets the bandwidth requirements of users [Bahnasse (2018)]. Our proposed algorithm 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.
4. Our proposed algorithm can efficiently and credible maximizing the resource utilization while ensures high admission for higher priority classes in a full network at same time which, as we articulate, cannot be sufficiently achieved by other existing schemes due to priority constraints.
5. Considering end-to-end delay as recommended for 5G systems, our proposed algorithm addresses the impacts of including end-to-end delay on the allocation process, and demonstrates affect the extend of maximizing overall resource utilization compared to other algorithms.
6. Suitable strategy for future application scenarios:
   Our solution can be deployable in any queue-based system. However, the benefits of our algorithm is more distinctive for future networks starting from 5G due to the massive bandwidth requirements with prioritization under limited network resources in congested and extreme scenarios such as emergency and disasters, especially under network slicing environment.

CHAPTER 2

Background and Literature Review

This chapter provides a study of the relevant technologies, techniques, and literature. It includes a discussion of the current state of the art in relation to the area we are targeting for our study, an introduction discussing the relationship between the various concepts and technologies, and then a detailed description of each proposed concept in the following subsection. Furthermore, we contrast our work with the existing literature and provide a rationale for our proposed approache. Last but not least, we present some of the work related to the thesis proposal.

2.1 Introduction

The new networking market is at an inflection point, 5G is being phased out, leading to another model of B5G with incredibly rigid QoS specifications identified by unprecedented bandwidth, computing capacities and ultra-reliable low latency communications (URLLC) [J. Sachs (2018), A. Karimi (2020)]. This upsets the conventional market models and architectures of the networking industry and the vertical market. Of course, several, if not most, of the network services of today will continue to play a vital role in the future. However, more improvements in their management are required in order to render them much more functional and cost-effective than before.

As a consequence, the various evolving network themes that are emerging in the immediate future, or have already arisen but have not yet been completely developed [Kunz (2019), L. U. Khan (2020)], put forth areas of research concern that both academia and business need to explore in order to face new problems that require strategies to leverage future network services (Network 2030 Services). In addition, owing to the rapid growth of modern technology paradigms such as IoT, there has been a sharp rise in the amount of digital data produced globally in recent years. According to the global supplier of knowledge IHS Markit would have 125 billion smart devices accessible by 2030 [Sharma (2019)], although Cisco expects that more than 11 billion cell devices will be paired by 2021 [Cisco (2018)]. Within a decade, the value of digital data exchanged has risen 200-fold by 2020 and this multiplication rate is predicted to rise by 1000-fold by 2030 [W. Xiang (2017)]. As a consequence, computer-intensive technologies and business structures have developed at a rapid speed, bringing remote cloud connectivity and processing architecture capacities to the limits. However, as Niels Bohr and others have pointed out in the past, "it is difficult to make predictions, especially about the future." [ClemmJNSM20 (2020)] Several predictions are indeed predictions, yet strongly confident that they will prove true given the pattern of the networking ecosystem, which is highly complex,

ever-growing and highly predictable.

Recently, ITU-T set up a Network Focus Group 2030 to research the set of network use cases anticipated in the next decade and to specify new network services and functions expected as a result [O. Brien (2019)]. Targeted new network networks impose new, strict management infrastructure requirements [Li (2018)]. Such usage cases are mainly defined by networks that have very specific timing specifications in terms of end-to-end latency (E2E), exceptionally high efficiency and high computing requirements. In the other side, early attempts against the 6G paradigm [A. Clemm3 (2020)] are seen as a catalyst towards its growth and development by illustrating particular issues such as the need for the E2E URLLC, but the need for the URLLC has also been set [H. Amirpour (2020)] under 5G.
Moreover, those use cases are highly time sensitive and need high computing resources from one or more sources to one or more destination nodes in an immersive manner over the network. Such as Holographic service representations entail massive volumes of data, and it is thus inevitable that completely dimensional 3D imagery would pose significant challenges to existing networks owing to the need for incredibly low latency requirements, huge improvements in bandwidth, and intense computing capacities marked by determinism and real-time specifications, stability and network versatility. The latency criteria is 10 ms [G. Liu (2020), Darlis (2017), A. Clemm4 (2020)] in order to enable instant viewer location change at approximately 60 frames per second. Present methods to network and service management are deemed ineffective since they are inadequate to serve simple networks with sufficiently low latency (in-time services with quantifiable latency), adequate bandwidth and heavy computing requirements. In comparison, current systems do not completely enable the idea of aggregated bandwidth, which is pooled and dynamically reallocated over a set of flows. That said, there is a need for an emerging network and service management approach (techniques) that will include network functionality to allow support for such kind of services to become a reality.

The 5G networking era played a significant role in seeing it as a solution to usage cases such as mixed reality, virtual and augmented reality, automated vehicles, and real-time apps, among others for standardization and deployment. However, the technical developments used by the 5G service should be used as feedback for the investigation of Network 2030 Networks in order to meet high-level device automation, self-awareness, flexibility, cost-effective activity and coordination [B. Blanco (2017)], with a view to the long-term successful delivery of next generation services.

With a view to overcoming the problem of low latency, extensive bandwidth and strong computing requirements imposed by 3GPP use cases, we envisage a service deployment algorithm that uses SKM intelligence to effectively allocate, manage,and control slice resources under several constraints in a multi-slice scenario, such as priority, bandwidth, and E2E delay with targeting to maximize the total resource usage in the substrate network, alongside other technologies such as NFV and SDN, in order to realize the promises of 5G technologies [ITUSDN3300 (2014)], [ITU-T T-REC-Y.3101 (2018)]. Great efforts are ongoing through many organizations, network operators and vendors, all collaborating in formulating the demanded standards to allow for a smooth combination between the

technologies. Many of the above principles are addressed in this chapter and how they apply to, or may be sought with, the investigation of a novel network management approach to help 3GPP use cases.

2.2 5G and Beyond 5G

The new trending network infrastructure of beyond 5G has forced conventional network management structures to experience innovative adjustments switching from the concept of linking all to real-time experiences. Therefore, a comprehensive comprehension of 5G and beyond 5G is paramount to suggest a durable system. Given the vast amount of networks and applications to be served in 5G and beyond 5G, consumer preferences are very homogeneous and close to the expectations of the previous technology, namely higher data speeds, low latency and cheaper services. However, future network generation is classified as 'Precision Smart Networking and Servicing Evolution Beyond 5G.' On one hand, the expansion of next generation networks would have a profound effect on the network management side to help the future usage cases. Many studies and documentations [A. Karimi (2020)] have listed the key discrepancies between the two technologies, but almost all of them appear to emerge from the latest proposed use cases that force the existing networks to fulfill their stringent QoS specifications, as seen in Fig. 2.1.
With new demands, network providers must revisit their market and technological models. This incentive provides an avenue for analysis. Examples of why beyond 5G is used include: HTC, Tactile Internet for remote operations (TIRO), Intelligent activity network (ION), Network and computing convergence (NCC), Space-terrestrial integrated network (STIN) and both of these have incredibly specific network specifications, because existing networks must adapt to satisfy these requirements. However, this would not disqualify the importance of 5G innovations such as Network Slicing, NFV, SDN, etc. They will be used and strengthened to satisfy the increasing network needs and promote potential network services.

2.3 5G use cases and different requirements

5G is attracting global attention not because it is a future generation of the network but because it would be able to cover use cases in addition to the traditional mobile broadband/communications. Namely, Third-Generation Partnership Project (3GPP) has defined four main slices types [3GPP1 (2020)] [3GPP2 (2020)]: the first slice type is targeted at ultra-high data rates (eMBB) as required for 4K or immersive 3d video, second slice type is specifically targeted for devices that require massive connections like agriculture (MIoT) and need efficiency. Besides, third slice type is targeted for ultra-low latency and high reliability services like self-driving vehicles (URRLC), and fourth slice type is targeted at advanced driving assistance services and needs ultra-low latency and high data rates (V2X). While the initial standards work calls for only three slice types, the architectures are adaptable to future slice types. These use cases come with a variety of technical requirements. The requirements become more complicated when one delves into each use case. In light of the

Figure 2.1: Key challenges in 5G and beyond 5G technologies [Albreem (2015)]

different requirements, mobile operators must have a way that is cost-effective and agile (i.e., fast time-to-market) to address the variety of requirements. Otherwise, mobile operators would have to extend several networks to satisfy the requirements of different customers including consumers using broadband and local factories that need mission-critical communications. Network slicing is a prospective candidate for addressing this challenge, as it allows a single physical network to perform many virtual networks that are optimized for various use cases. The potential of network slicing is also common in 3GPP's standardization activities, where network slicing is a vital feature of 5GC (5G Core). Network slicing is no longer a choice but rather an indispensable component to fully exploit the potential of 5G networks.

2.4 Network slicing

Network slicing is a novel network paradigm developed within the context of recent 5G networks which proposes the partition of the physical network infrastructure into multiple self-contained logical

(or virtual) networks called slices as shown in Fig. 2.2. More formally, based on the descriptions of 3GPP [3GPP (2019)], a Network Slice is a logical network that offers specific network capabilities and network characteristics. Even more, a Network Slice Instance is described as a set of Network Function Instances and the needed physical resources which compose a deployed Network Slice. During the last years, the network slicing approach has consolidated as the main enabler for 5G networks. As explained in [Katsutoshi Kusume (2015)], network slices leverage deploying services with contradictory demands across a shared infrastructure, facilitating the management of the network. In particular, slicing the network provides to independently configure the networks edge-to-edge and describe specific functions for every case, while sharing the same infrastructure. Despite their end-to-end existence, and the inherent purpose of slices being implemented to provide such facilities, viewed from the point of view of an Access Point (AP), we consider a slice as a collection of traffic flows with some typical characteristics, and demanding certain output criteria. Complementary to the previous definition, and, as a working concept of the study in this thesis. Under this perspective, several slice instances include the same sort of system as the source or destination, the flows of a VoIP service, the flows from a customer with a specified carrier, etc. Furthermore, a slice will help flows from different clients, while at the same time, a client can engage in multiple slices. However, flow does not belong to a particular slice, but is independent between any other slice. As stated, this viewpoint of a network slice is complementary to current concepts and will aid in the design of a network slicing solution. In network slicing, infrastructure vendors lend their network expertise to emerging market players, such as virtual mobile network operators (VMNOs), and OTT networks. These novel players have the ability to grant access to the presented capital. Ensuring the effective capital utilization and consistency separation are the greatest obstacles confronting this paradigm. A key slice requirement is defined in [3GPP (2018)] in which two forms of slice specifications are denoted. Slices can be customized to include various functional specifications, such as priority, charging, or flexibility. For example, a slice may be configured to provide only the functions that are required for the particular service, thus removing unnecessary functions.

- Different types of slices can include different specifications such as latency or durability. However, several other alternatives to the issue of network slicing have already been suggested. During the work we've suggested to divide slicing into two distinct varieties:
- Quality-of-Service Slicing (QoSS): Slices that guarantee the Quality of Service even though the services are inadequate. An definition of a slice consists of ensuring a minimum delay or a minimum bandwidth for a specified operation.
- Infrastructure-Sharing Slicing (ISS): analogous to the concept of network virtualization, a number of services are reserved for the sole usage of a tenant. The occupant has full influence over the network facilities and network operations situated within their slice.

Thus, the QoSS variant allows the slice provider to specify output goals for each flow in the slice, while the ISS variant requires a tenant to indicate a collection of resources to be reserved for the entire slice. Although the ISS method is not expressly included in the previous concepts it offers

neither output nor practical guarantees, but we envision it has potential to be used in certain situations, as can be shown in recent experiments which suggest a similar approach [Sarkar (2017), Coronado (2018), Jorgensen (2019)].

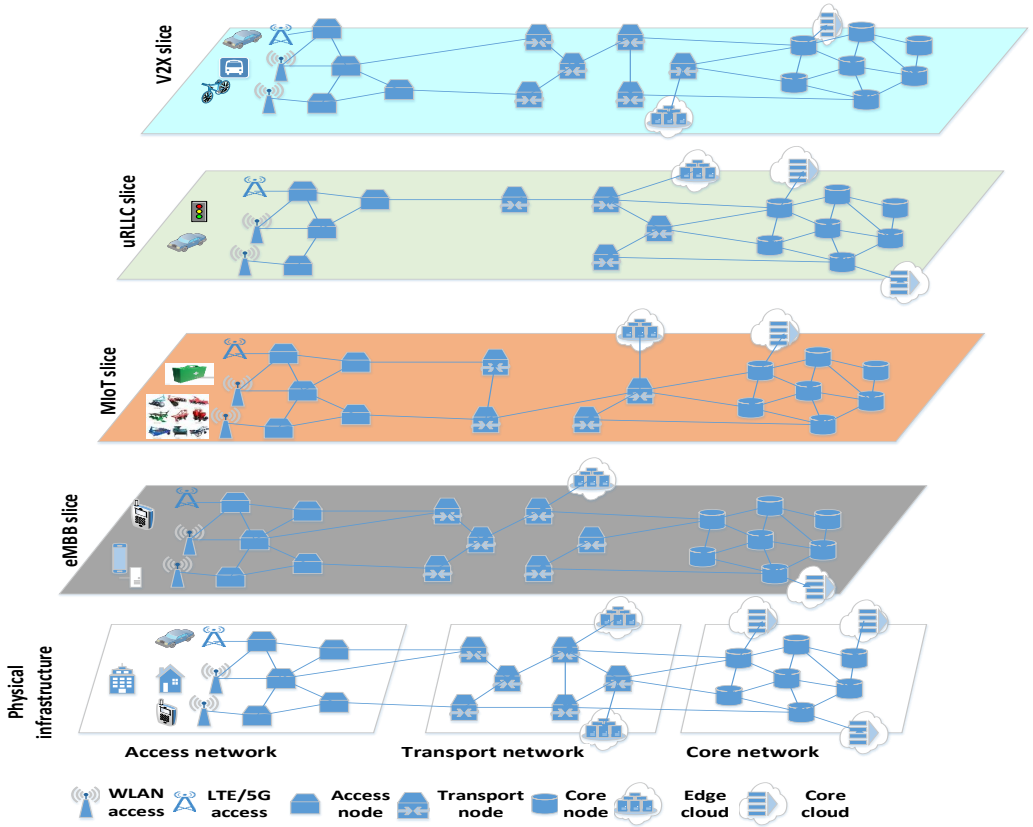


Figure 2.2: E2E 5G network slicing architecture

2.4.1 Heterogeneous service differentiation

In the exist context, when there are a multiplicity of services and equipment that wireless networks have to compete with, slicing becomes a means to separate and achieve different requirements simultaneously. On distributing resources, slicing enables us to build personalized services with fine control of aspects along with Quality of Service [Derakhshani (2015)]. The concept is to split the network into distinct subsets of resources and capacities to build a dynamic range of services targeted to various usage cases. Moreover, in [Ericsson (2015)], slicing is regarded as one of the primary enablers of potential 5G Systems required to handle the anticipated heterogeneous requirements. In addition, it is possible to specify unique slices per application, each of which can include advanced networking [Heming Wen (2013)]. There is a layer of abstraction over the slice so as to monitor the network as a black box to conveniently define application requirements. Another alternative solution is to provide personalized prototypes for each form of system or consumer criteria. Network slices are effective since they can be tailored by a single client and without regard to available resources. This dynamism is the main distinction from related ideas as well as virtual private networks (VPNs).

2.4.2 Network management

In [Katsutoshi Kusume (2015)], the author explores how to handle diverse applications with diverging specifications on a shared infrastructure. Slicing the network enables to uniquely customize the slices edge-to-edge and specify separate roles for each event, while sharing the same infrastructure. For example, slicing enables to assign only the required tasks, and it's versatile enough to satisfy a wide variety of communication demands. Slicing often allows ability to automatically build and destroy slices based on the operators strategy, using Network Feature Virtualization (NFV) and Software Driven Networks (SDN). The aim is to virtualize any mission-critical functions and program all remaining activities to be configurable [Javan Erfanian (2015)]. Even further, in the case of slices specified per form of service or unit, it's understood which service each slice is servicing and hence can be streamlined by eliminating functions that are not required. In the case of a slice allowing access to static sensors, accessibility control may be reduced to a minimal. This way, management becomes streamlined, integrating autonomous management with each particular slice.

2.4.3 Infrastructure sharing

Often, another incentive for slicing is sharing infrastructure. It is analogous to the service separation principle but, in this situation, each slice will be used by a different provider providing its own services. Also, in 2014 there were in the UK 41 mobile voice over internet protocol (VoIP) providers, who are clients of mobile infrastructure providers [Ofcom (2014)]. Many of these firms have voice, SMS, and data networks similarly to the existing provider as well. For infrastructure maintenance, slicing allows attain reliable and productive operations. On a particular point of view, the concept of exchanging the networks allows operators more freedom to adjust their conceptual network and better utilize their resources [Zaki (2012)]. This presumption is also supported by the Telemanagement Forum [Graham (2015)], which also stated that 5G would provide several virtual networks through a common infrastructure.

2.4.4 Flexibility for new services and business models

From a market viewpoint, network slicing facilitates the implementation of new usage cases without raising costs since each slice needs a separate piece of infrastructure. This will allow to offer coverage to devices with low traffic demands on high population areas (e.g., IoT), without raising prices, which 5G does. Besides, as a simplified API for programming, slicing leverages the Anything as a Service (XaaS) business model and helps third parties to discover different opportunities.

2.5 Network slicing enablers

In this section the principles of NFV and SDN are briefly explained and several recent works reviewing these subjects are reviewed. These principles are fundamental for the intent of network slicing.

2.5.1 Software-defined networking

SDN decouples the control plane that determines the traffic from the data plane which physically manages the traffic based on the configurations given from the control [Bhattacharjee (1997), Kreutz (2015)]. These strategies include a range of techniques allowing network operators to "directly program, orchestrate, control and manage" network infrastructure to enable the design, distribution and execution of network services. This is achieved in SDN by transferring network resources to a dedicated network element called the controller that offers an abstracted view of the network, which also provides frameworks to monitor and maintain the network [S. Agarwal (2013)].

2.5.2 SDN architecture

To remain inside the suggested standard architectures, and not distract our investigation from specifications established by industry standards bodies, we present an SDN architecture from an organisation dedicated towards the implementation and standardization of SDN known as Open Networking Foundation (ONF) [ONF (2012), ONF (2014)] which provides a schematic representation of the standard SDN architecture shown in Fig. 2.3. The design is focused on the core objective of SDN i.e. separating the control plane and data plane and how they connect with each other through northbound and southbound accessible interfaces. SDN architecture is based on four logical layers:

1. The application layer: consists of a list of applications and services used by network users.
2. The data layer: consists of the network devices responsible for information forwarding.
3. The control layer: typically, a Network Operating System (NOS) allowing an abstraction suite (application programming interface) to access the resources (such as routing table, forwarding, and calculations) of a network node. It decides on routing, traffic engineering, and fault detection.
4. The management layer: This layer is responsible for arranging and supervising network devices (such as routers, servers, and switches) the management layer and control layer are often implemented on the same processor.

The interfaces are designed to connect two layers. Since we describe the hardware elements at the bottom and the applications and services at the top, the interfaces are often called the "north" and "south".

The North interface aims to pass on information from applications so that the controller can create a request that meets the application's Service Level Agreement (SLA).

The south interface allows configuration commands to be delivered to the data layer and raises network statistics from the latter to the control layer. Several protocols can be used in this interface such as OpenFlow (OF) [McKeown N (2008)], Network Configuration Protocol (NetConf) [M. Dallaglio (2016)], Open vSwitch Database (OVSDB) [B. Pfaff (2013)], and Simple Network Management Protocol (SNMP) [M. MacFaden (2003)]. In the case of a hybrid network (containing both recent SDN nodes and legacy nodes), two or more protocols can be used together, such as OpenFlow and NetConf. Fig. 2.3 illustrates an overview of the SDN architecture.

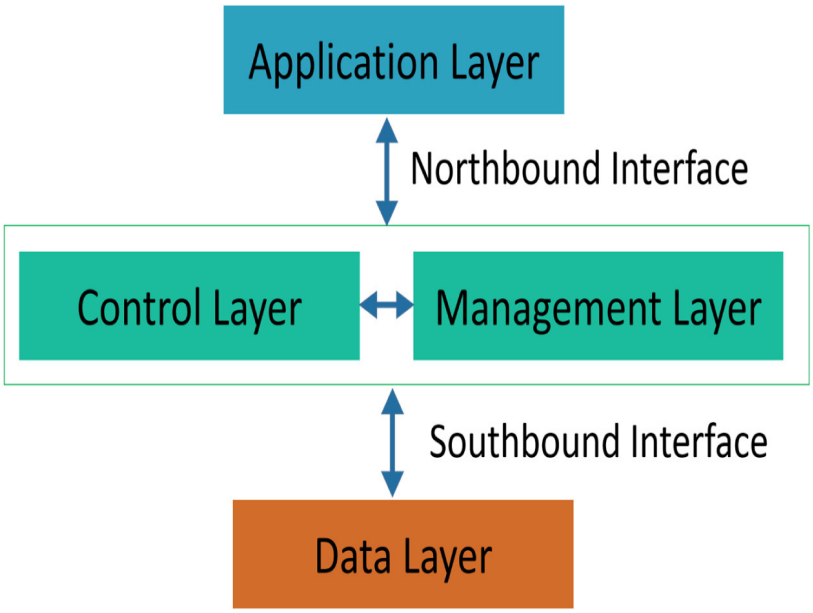


Figure 2.3: SDN architecture [Bahnasse (2018)]

2.5.3 Future potential of SDN

SDN is majorly used to monitor the forwarding of data currently, potential versions of the general definition could be a key enabler for network slicing due to any of the following reasons; 1) SDN abstracts away complicated management functions from the customer, implying that, the control plane may be responsible for the placement and orchestration of services. End users may only decide what service they request, and the decision where to instantiate it can be taken care of by the control plane policies, 2) the controller as global view may be leveraged for service exploration and to gather measurement data on the state of the network, and 3) the capacity of SDN to dynamically reconfigure the network proves crucial in complex edge environments for delay-sensitive application.

Deploying a sliced network may be very complicated if not implemented properly. In our view, SDN is essential for ensuring these activities are quickly achieved and increasing the reliability and programmability a sliced network would need. SDN is, however, not the solution to network slicing itself, but is instead listed as an illustration of several possible solutions.

With new strict 5G services, where consumer mobility and shifts in service demands is strong, the sporadic access to unstable computing nodes, SDN-enabled networks may push new flow laws to the network so as to route traffic as planned. SDN may provide assurances on the level of service provided by edge components, such as dynamically balancing the computing resources handled by kubernetes with the network resources demanded for by service requests. Few reports have already explored the similarity of SDN ideas to edge conditions. Heuschkel et al. [Heuschkel (2017)] present a protocol which goes beyond the core network and enables it to spread to end devices. Bi et al. [Bi (2018)] demonstrate how the regulation of versatility can be decoupled from the forwarding of data

and obtains with SDN. [Barakabitze (2020)] examines whether SDN can be used to benefit from network slicing. However, researchers have not tested the methodology in future network facilities, especially for 3GPP use cases.

2.5.4 Network functions virtualization (NFV)

In contrast to SDN, NFV abstracts the network functions (including the network forwarding and control functions) from the hardware [ETSI (2013c)] [ETSI (2014a)]. That is, the functions of the traditionally dedicated network equipment (e.g., router, firewall and load balancers) can be provided as software functions running on virtual machines. NFV enables savings in both capital expenditures and operating expenses as dedicated hardware can run on standard commodity servers. Furthermore, there is no need to over provision data or service centers as server capacity can be changed on demand through the software settings [ETSI (2014b)]. This is especially relevant for mobile networks, where there is a variety of proprietary hardware, which is expensive to operate on which it is challenging to launch new services. Therefore, network services based on VNFs can be put on the market, maintained and upgraded more easily and timely which creates a great potential for increasing the usage value of network resources.

2.5.5 NFV architecture

The NFV architecture introduced by the European Telecommunications Standards Institute (ETSI) is promoting to describe standards for NFV implementation. Each element of the architecture is based on these standards to support better stability and interoperability [ETSI (2013a)]. NFV architecture consists of: VNFs can be deployed and reassigned to share different physical and virtual resources of the infrastructure, in order to guarantee scalability and performance requirements. This allows the service providers to deploy new and flexible services more rapidly. In general, there are three main components in the NFV architecture as shown in Fig. 2.4: VNFs, Network Function Virtualization Infrastructure (NFVI) and the NFV Management and Orchestrating (NFV-MANO). These are described below:

- **VNFs:** are software implementations of network functions that can be deployed on a network functions virtualization infrastructure (NFVI) [ETSI (2013a)].
- **NFVI:** is the totality of all hardware and software components that build the environment where VNFs are deployed. The NFV infrastructure can span several locations, e.g. datacenters and public or private hybrid clouds. The network providing connectivity between these locations is considered as part of the NFVI. Physical resources typically include computing, storage and connectivity for VNFs through virtualization layer that abstracts the physical resources [ETSI (2013a)] [ETSI (2013c)].
- **NFV-MANO:** is the collection of functional blocks, data repositories through which these blocks, and reference points and interfaces through which these functional blocks exchange information for the purpose of managing and orchestrating NFVI and VNFs [ETSI (2013a)].

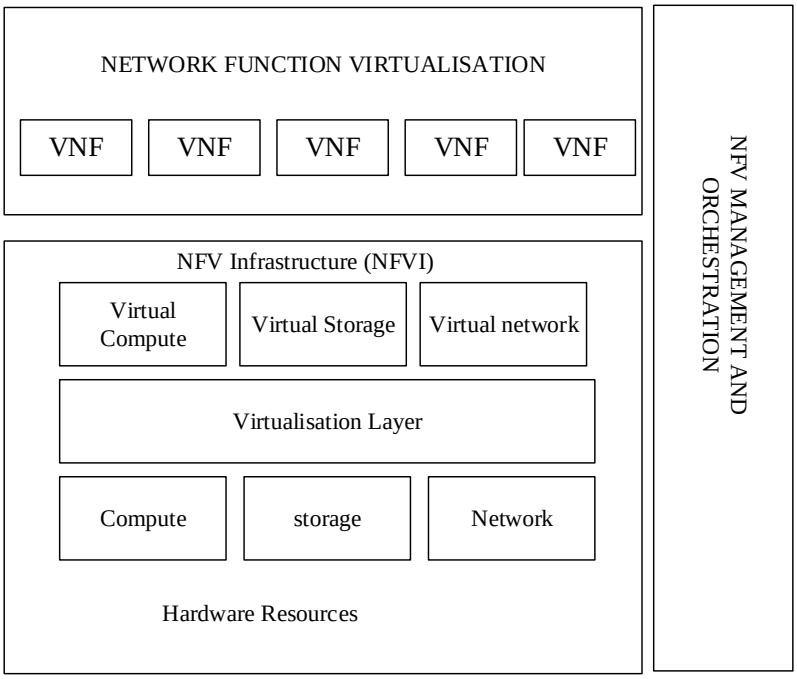


Figure 2.4: High-level NFV framework

2.5.6 Future potential of NFV

Future forecasts without the usage of NFV technologies cannot be produced [Mohammadkhan (2020)]. The key advantage of NFV to network operators and service providers is that it makes network implementations and activity considerably more cost-effective [Hawilo (2014)]. The ongoing transition to virtual networks is also interesting because of recent network technologies. As Abdelwahab et al. [Abdelwahab (2016)] indicates, NFV has a number of future market prospects, and there is a lot of interest in NFV as a business model. We see NFV as enabling the slicing of a cellular network. It will be simpler to build and handle data slices if the data could be taken from proprietary hardware and virtualized, and then centrally handled. In the following, we explain several examples of how network slicing plans may be extended to network protection.

With NFV, service providers can run network functions on standard hardware instead of dedicated hardware. Also, because network functions are virtualized, multiple functions can be run on a single server. This means that less physical hardware is needed, which provides for resource consolidation that results in physical space, power, and overall cost reductions. NFV gives providers the flexibility to run VNFs over different servers or move them around as needed when demand changes. This flexibility lets service providers offer services and applications faster [Herrera and Botero (2016)] [Mijumbi (2016)]. For example, if a customer demands a new network function, they can turn up a new VM to manage that demand. If the function is no longer needed, the VM can be decommissioned. This can also be a low-risk way to test the value of potential new service.

2.5.7 SDN, NFV, network slicing and 5G

In principle, SDN gives "flexible forwarding and steering of traffic in a physical or virtual network environment" whereas NFV gives "flexible placement of virtualized network functions across the network and cloud" [Obraczka (2016)]. These two complements each other to perform a truly programmable network. By decoupling the software and the hardware, and then the control plane of the data plane, the network can leverage commercial off-the-shelf (COTS) hardware while profiting from centralized control and simplified network nodes. Implementing SDN and NFV to the context of mobile networks enables network slicing. Indeed, the 5G core networks are designed by network slicing in mind and allow the separation of the control plane from the user plane (data forwarding) on top of the virtualized infrastructure. As the majority of user-plane traffic needs only very simple processing, it can run on low-cost hardware. Nevertheless, distributed control-plane entities need advanced processing. This allows cost-efficient scaling depending on the user plane demand, as the control-plane is independent of the user-plane [Basilier (2016)]. 5G PPP gives an overall architecture for 5G mobile networks that utilize network slicing [5GPPP (2017)]. In this architecture, networks slices are performed at the network level to support each individual service, and the network slices are programmable with programmable control. A virtualized infrastructure underlies in the resources and functional level. The slices are end-to-end, where they span from the access networks to the core networks and are controlled and orchestrated by the "secure network and service management" and "end-to-end secure service orchestrator." There are, of course, fundamental difficulties that are included with SDN and NFV. First, whereas traditional mobile networks consist of physical devices coupled with specific functionality, which makes deployment and operations modular, network slicing introduces layers of complexity. For instance, mobile operators could expand specific devices in necessary locations (e.g., controllers of base stations in densely populated areas) according to demand, but it is more complex conceptually to use and operate commodity hardware in various places let alone the instantiation/termination of the required software functions. Second, the softwarization of networks is from the Information Technology (IT) industry where there is a high concentrated server farm attempting to process data efficiently. The Communications Technology (CT) industry, nevertheless, has a fundamentally various objective as its focus is more on the transport of the data than the processing of the data. Trying to fit the IT industry's software paradigm into the CT industry's mission-critical data transport paradigm (also known as five 9's: 99.999% reliability) is a hard challenge in itself.

2.6 Interpretation of the network slicing problem

This section contains more information about the dissertation's various issues and overviews some of the research principles and methods. However, all the problems listed are highly dependent on the network slicing model and the strategies used to allocate resources as well as guarantee high level of QoS for multiple 5G services.

2.6.1 Modeling demands and resources

One of the problems to solve in order to incorporate network slicing is to identify the services which would be allocated to various slices and how a tenant demands a slice.

Moreover, because it would be far too costly to allocate a complete end-to-end network to each type of slice, the network infrastructure that promotes 5G will employ sharing techniques (SDN and virtualization technologies such as NFV), which allow for multiple slice types to coexist without having too many resources. In [Ericsson (2015)] slicing is introduced as one of the key capabilities of 5G to handle the expected heterogeneous demands of future mobile networks. Furthermore, Network slicing is still nascent and needs to manage and orchestration of Virtual Network Functions (VNFs), mapping and service descriptions. In existing literature, little works are conducted for deployment of end-to-end (E2E) network slicing although it is necessary for the realization of network slices which provide the operators with the ability to customize networks meeting various service demands. Also, E2E slices need to instantiated rapidly. Moreover, most of these works are addressing core network resource slicing and there exist few works focused on link resource management with prioritization on traffic classes which offer new research challenges such as bandwidth allocation/reservation along with links of a requested path and isolation between traffic classes [M. Jiang (2016)]- [A. Huang (2020)]. These works focused on network slicing from a bandwidth aspect since future network technologies are characterized by extremely wide bandwidth requirements that will be accessible by users under limited available bandwidth resources. However, these works still suffering from inefficient in terms of deployment and management of end-to-end slices and need crucial and promising models to address the above challenges [M. Jiang (2016)]. Table.2.1 shows some studies that dealt with bandwidth resources in network slicing.

Table 2.1: Studies on the bandwidth resource in network slicing

Paper	Summary
[M. Jiang (2016)]	Presented a novel heuristic based admission control mechanism able to dynamically allocate network bandwidth resources to different slices.
[H. Zhang (2017)]	Presented a scheme for managing mobility between different access networks based on network slicing.
[S. Xiao (2018)]	Focused on the dynamic resource allocation problem of bandwidth in transport network slices.
[C. Suzhi (2019)]	Focused on allocating bandwidth resources to different slices that can be performed according to the service QoS.
[J. Li (2019)]	Proposed a bandwidth slicing mechanism, in which the bandwidth can be provisioned effectively to meet their different delay requirements.
[H. Uzawa (2020)]	Proposed a dynamic bandwidth allocation scheme for network slicing that performs uplink bandwidth allocations in a different manner.
[L. Feng (2020)]	Focused on guaranteeing the latency and reliability of sporadic uRLLC and eMBB uplink traffic.
[T. V. K. Buyakar (2020)]	Proposed an algorithm that supports QoS parameters, including Guaranteed Bit Rate (GBR) and Maximum Delay Budget.
[A. Huang (2020)]	Focused on the distributed network slicing utilizing the spectrum resource.

we will introduce some state of the art work on QoS and resource allocation to improve the programmability and flexibility of networks and discuss the features of the proposed solutions. We then compare them with the proposed framework based on the identified features. To better compare these solutions, we categorize them into three groups, namely QoS solutions based on SDN, QoS solutions to support specific network applications, and QoS solutions for network slicing in 5G.

2.6.2 Technical overview of QoS

The fundamental objective of any QoS algorithm is to ensure that excessive congestion does not occur for the demands with assured QoS. During the past several years, numerous QoS management models have been broadly studied and described for instance Best Effort (BE) [RFC 2474], Integrated Services (IntServ), Internet Engineering Task Force (IETF) [RFC 1633] and DiffServ [RFC 2475] were broadly analyzed and implemented. These models based on the specific use of the octet named traffic class [RFC 2460].

DiffServ model aims at solving the limitations and problems of IntServ and BE management models even in the congested network case. This is achieved by introducing three key operation primitives: (i) Definition of local service policies at each router (the so-called Per-Hop Behavior or PHB), (ii) Utilization of loose resource reservations for traffic classes, and (iii) Flexible traffic class identification mechanism based on three main classes plus class prioritization. However, DiffServ model is unable to ensure end-to-end QoS levels by its own, since no traffic management is supported. At this point, MPLS-TE attracted much attention [RFC 2702, RFC 3272].

Thus, DS-MPLS networks using their TE capabilities allow guarantee of QoS for each type of traffic according to the class of service it belongs to [RFC 3564]. It ensures the management and allocation of available bandwidth in the network. The benefits of the class of service constraints are to maintain the appropriate QoS for the required bandwidth. One of the key algorithms of the DS-TE is the specification of a bandwidth constraint model, which describes the allocation of the bandwidth to individual class types in order to enhance the QoS of traffic streams and to optimize resource utilization as described in [RFC 2702].

In general, it should be ensured that some network resources do not become over utilized and congested while other subsets along alternate paths remain underutilized. Bandwidth is a crucial resource in contemporary networks. Therefore, advanced techniques for bandwidth resource allocation and management are required.

2.6.3 Existing bandwidth resource allocation and QoS models

Several works in the literature dealt with the dynamic bandwidth allocation for guaranteeing a given QoS level per class and optimizing utilization. Preemption and squatting are consistent approaches that can be adapted to guarantee QoS. Thus, BAMs such as MAM, RDM, and AllocTC, with a reservation are used as preemption strategies while BAM with squatting and kicking strategies (soft and hard) are discussed in [Hesselbach (2016)].

In [Sadon (2012)], the authors proposed a new algorithm based on RDM to support dynamic bandwidth allocation for DiffServ classes and improve bandwidth efficiency by allowing the triple-play services to share the bandwidth. The allocation of bandwidth is based on the classification and prioritization of service. The proposed scheme is applied for Ethernet Passive Optical Network (EPON) and provides fairness factor and services priority for the required bandwidth of the request.

The general problem of the algorithms based on RDM is that the resources reservation is carried out from the bottom to top, which means that lower priority classes share its resources with higher ones and not the inverse. Also, the general problems of the algorithms based on MAM are that any class cannot use the available resources from another given class. In order to overcome the problems of MAM and RDM performance, several works have been carried out proposing new dynamic bandwidth sharing algorithms based on modified MAM or RDM strategies such as [Adami (2007), Tata (2013), Trivisonno (2015), Dantas (2014), Subhashini (2015), Neto (2008)]. However, these models can not guarantee high admission for higher priority classes and give 100% network utilization at the same time.

Efficient utilization can be achieved by making the reservation of resources either from the top or down. In this regards, the authors in [R.F. Reale (2011)], proposed a model called AllocTC, which provides sharing of the unusable bandwidth of high bandwidth applications priority with low priority and vice versa. In [Dures (2017)], the authors studied the behaviour and resource allocation characteristics of the BAMs then they compared distinct BAMs using different traffic scenarios in order to investigate the impact of a dynamic change of the BAM configured in the EON network. The authors prove by simulation that AllocTC is more efficient in terms of optimizing the utilization of the link and that it is better suited for elastic traffic and high bandwidth utilization. The authors in [R.F. Reale (2014)], propose a new approach with a combination of (MAM, RDM, G-RDM, and AllocTC) models based on a controller by using different metrics to switch from one model to another one in order to improve performance in terms of link utilization, blocking probability, and packet number.

Since SDN has been viewed as a promising network technology for 5G, SDN-based QoS issues have equally received much attention. Commonly, there is a function module of QoS in the SDN controller to implement network resource monitoring and scheduling. For example, Tomovic et al. presented a controller framework with QoS provisioning for multimedia applications [Tomovic (2014)]. In this framework, four key function blocks (i.e., resource monitoring, route calculation, call admission control and resource reservation) were integrated into the controller to perform QoS management. Dutra et al. [Dutra (2017)] provided operators to assign network resources through the feature of the queue in OpenFlow so that over-provisioning of bandwidth resources can be decreased or eliminated. Pan et al. presented a programmable packet scheduling framework OpenSched [T. Pan (2017)], which was a layered architecture to glue the QoS applications, the controller and the switches together, considering flexible northbound interface, controller-switch interaction and effective southbound protocol handling, as well as QoS policy execution at the switch side. A prototype based on ONOS and OVS demonstrated that it can facilitate flexible network resource provisioning. Oliveira et al. [Oliveira (2018)] proposed a QoS provisioning architecture to support the classification of services

and negotiation of QoS demands among applications and the SDN controller, which can control and improve network performance on-demand and in a timely fashion. In [Bahnasse (2018)], the authors proposed a new SDN-based architecture following a new smart and dynamic model (smart Alloc) for allocation and managing the QoS and routing with QoS constraints for a DS-TE network. This model is based on RDM and AllocTC strategies and aims, firstly, to classify 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 calculate the fairness index in order to allocate resources precisely to all flows 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, all these models cannot guarantee high admission for higher priority classes. Table.2.2 Shows some studies that dealt with bandwidth resources based on BAMs.

2.6.4 QoS solutions to support specific network applications

Some research works concentrate on the QoS solutions for specific network applications, such as cloud datacenter network, smart grid network, energy network and remote medical network. Tajiki et al. studied QoS optimization with the smallest network reconfiguration expenses in the cloud datacenter [Tajiki (2017)]. A forwarding table compression technique was created to perform resource reallocation, which can be deployed as an application module in the SDN controller. The analysis results showed that it efficiently decreased the network reconfiguration overhead while meeting the QoS demands. In the work of [Rezaee (2020)] the authors introduced a QoS model based on SDN for the smart grid network. In this model, a content-aware queuing algorithm was devised so that traffic flows were classified into various groups, which finally gave low latency connection for smart grid network. Qiu et al. [Qiu (2019)] The advantage of this algorithm were to solve the problem of the interaction between multiple controllers using Artificial Intelligence (AI) technology so that they can automatically negotiate QoS parameters. A QoS-sensitive application for medical systems is presented in [Venkatesh (2019)]. The authors offered a multi-path routing algorithm to ensure QoS requirements and optimize the QoS of medical information transmission in an OpenStack environment using the OpenContrail controller.

2.6.5 QoS solutions for network slicing in 5G

Currently, there are also some research works concentrating on QoS to maintain network slicing in 5G. For instance, Rafael et al. studied the Quality of Experience/Quality of Service (QoE/QoS) of 5G-enabled optical networks [Montero (2019)], which concentrated on the E2E service delivery. An architecture of network slicing provisioning with QoS guarantee was introduced, promoting 5G service chaining in cross-domain optical networks. A policy-based monitoring and actuation framework was used to support the desired QoS demands for E2E network slice. Nevertheless, this framework did not provide the cooperation mechanism among SDN controllers and NFV entities to perform QoS decision in the context of network slicing when the network topology modified. A.

Table 2.2: Studies on the bandwidth resource based on BAMs

Paper	Summary
[Hesselbach (2016)]	Proposes an allocation approach in EON that focuses on using squatting and kicking techniques based on RDM modification to maximize the overall use of channels, and allocate resources more efficiently.
[Sadon (2012)]	Demonstrates a new hierarchical Dynamic Bandwidth Allocation algorithm using the RDM to allocate bandwidth for ONU in an EPON.
[Adami (2007)]	Introduces a novel bandwidth constraints algorithm, called G-RDM.
[Tata (2013)]	Proposes a new Bandwidth Constraints Model for MPLS networks, called CAM (Courteous Allocation Model).
[Trivisonno (2015)]	Introduces and examines three alternative NRM methods: Full Sharing, Full Split and Russian Dolls.
[Dantas (2014)]	Propose a differentiated service methodology that implements constrained resources allocation according to demand's priority level for WDM networks.
[Subhashini (2015)]	Demonstrates a new user prioritized dynamic bandwidth allocation algorithm.
[Neto (2008)]	Proposes the ADAPT-RDM algorithm, suitable to allocation or not of LSPs according DiffServ-aware MPLS Traffic Engineering and with the utilization of the RDM bandwidth constraint model.
[R.F. Reale (2011)]	A new bandwidth allocation model for DS-TE networks is presented.
[Dures (2017)]	Evaluates the applicability of bandwidth allocation models for EON slot allocation.
[R.F. Reale (2014)]	A Generalized Bandwidth Allocation Model (G-BAM) for IP/MPLS/DS-TE Networks is presented.
[Tomovic (2014)]	Presents a new SDN control framework for QoS provisioning. Beside QoS provisioning for priority flows, the proposed solution aims at minimizing degradation of best-effort traffic.
[Dutra (2017)]	proposes a solution that enables the E2E-QoS based on the queue support in OpenFlow, allowing an operator with a SDN-enabled network to efficiently allocate the network resources according to the users' demands, reducing or even eliminating the need for over-provisioning.
[T. Pan (2017)]	Suggests a layered architecture that glues the QoS apps, the controller and the switches together to maximally unleash the power of centralized QoS control.
[Oliveira (2018)]	proposes a QoS provision architecture exploiting the capabilities of SDN. The approach allows the specification of classes of service and also negotiates the QoS requirements between applications and the SDN network controller.
[Bahnasse (2018)]	Proposes a novel SDN architecture for smart MPLS DS-TE Management, which aims to dynamically manage the bandwidth and to ensure the segment routing.

Sgambelluri et al. [Sgambelluri (2019)] In this solution, a stateful backward recursive path procedure was used to support the E2E connection services. Analysis results confirmed that this solution can promote the automatic establishment of QoS-based E2E connection across multi-operator network domains. Nevertheless, this orchestration system was not elastic enough to maintain the scalability for the advertisement of resources and dynamic connection services. Vincenzi et al. [Vincenzi (2017)] presented a thorough discussion of the challenges that network slicing effects in the various network parts and designed a collaborative game to study the potential interaction aspects between the participants. Sattar et al. [Sattar (2019)] addressed the question of optimal allocation of a slice in

5G core networks by tackling two difficulties, namely function isolation and guaranteeing end-to-end delay for a slice. Nevertheless, SDN and NFV technologies were not applied to these solutions.

2.6.6 Adaptive QoS assignment for multi-path 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 [Wu1 (2016), Wu2 (2016)]. The impact of multi-path and QoS evil 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 notably when demands are delicate to SLA restrictions. In [Quang (2018)], 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. While the authors in [Eramo (2018)] propose and evaluate the performance of an algorithm to allocate and compute optical bandwidth resources in an NFV environment so as to minimize their costs and by taking into account the different costs of the the Infrastructure Providers. In [Kuo (2018)] 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. In [Zsa (2001)] the authors introduced a global path optimization algorithm which includes two main parts: demand sorting and path allocation. The authors compute the priority of every demand as its bandwidth requirement distributed by its hop. The algorithm processes requests one by one in decreasing order of priorities. Before arranging requests, the algorithm updates the weight for every link, which represents the bandwidth usage. Moreover, for every demand, the least weighted path is chosen. Table.2.3 Shows some studies that dealt with path selection that taking into account how the resources are accessed.

Table 2.3: Studies on the path selection that taking into account how the resources are accessed

Paper	Summary
[Wu1 (2016)]	Investigates the relationship between energy and goodput for real-time multimedia transmission with stream control transmission protocol over heterogeneous wireless networks.
[Wu2 (2016)]	Proposes an energy-video aware multipath transport protocol.
[Quang (2018)]	Provides a model of the adaptive and dynamic VNF allocation problem considering VNF migration.
[Eramo (2018)]	Proposes and to evaluates the performance of an algorithm for the computing and optical bandwidth resource allocation in NFV environment.
[Kuo (2018)]	Computes a proper routing path length, and decide, for each VNF in the service chain.
[Zsa (2001)]	Proposes an algorithm to solve the optimization of label switched paths (LSPs) in MPLS networks.

2.6.7 Bandwidth and time constraints

Regarding related work about bandwidth and delay constraints routing for network slicing, several algorithms have been proposed in the literature. In [Wang (1996)], a simple solution called a minimum delay algorithm has been proposed, which removes all links with insufficient residual bandwidth along the required paths and then determines the optimal path which has the least delay in the network. Based on the rule of least interference [Kodialam (2000)] used in the bandwidth restriction routing problem, an alternative algorithm called the delay-weighted maximum capacity routing algorithm [Yang (2001)] interprets the concept of interference differently when delay constraints are accepted. The algorithm calculates the delay-related shortest paths for all source and destination node pairs. The bottleneck links for these pathways were defined as critical, and the level of criticality of the link is a weight function related to the number of source and destination node pairs that are essential to it. By employing the extended Dijkstra shortest path algorithm [Chen (1999)], the lowest weight path that satisfies the delay and bandwidth constraints is chosen for each request. The authors in [Tomovic (2016)] believed that only looking at the bottleneck links on the lowest delay paths for all source and destination node pairs was not sufficient. Therefore, Yen's algorithm [Yen (1971)] is employed for each source and destination node pairs to calculate the candidate path set, including k loop-less paths in non-decreasing order of delay. Moreover, the weight function specified in the links is adjusted accordingly. Likewise, the experimental results of the least interference routing algorithm for bandwidth-restricted routing, although the delay-weighted maximum amplitude routing algorithm can yield good results Productivity and blocking ratio on small-scale networks, the running time becomes unacceptable as the size of the network increases. Due to the vast time complexity of Yen's algorithm which is employed to the k-shortest path problem, the running time of the proposed algorithm in [Tomovic (2016)] is much higher compared to the highest delay-weighted capacity routing algorithm.

Different from these works, in terms of path election, our proposal calculates an optimal routing path considering how the resources are accessed according to a predefined set of slices of service on multi-slice networks while considering E2E delay constraints. Our model performs stricter on priorities and significantly differentiates priorities under congested scenarios to optimize the utilization and provide high admission for higher priority slice users in terms of traffic, which is essential for the quality of service guarantee and SLA. Table.2.4 Shows some studies that dealt with bandwidth and delay constraints.

2.6.8 QoS metrics

QoS metrics are used to assess the impact of the allocating process on the services delivered to end-users by the physical network. Typical examples are path length, stress level, utilization, throughput, delay, jitter, and packet loss. Path length, it is a measure of the number of links between two physical nodes that are mapped to two interconnected physical nodes, the longer the path, the more resources required to perform the allocating. Stress level, representing how much loaded is a certain physical network resource when many physical nodes are mapped on it, causing the delay, limitation on

Table 2.4: Studies on the bandwidth and delay constraints

Paper	Summary
[Wang (1996)]	Examines the basic problem of QoS routing, namely, finding a path that satisfies multiple constraints, and its implications on routing metric selection.
[Kodialam (2000)]	Presents a new algorithm for dynamic routing of bandwidth-guaranteed tunnels when tunnel routing requests arrive one-by-one and there is no a priori knowledge regarding future requests.
[Yang (2001)]	Presents bandwidth-delay constrained routing algorithms that use knowledge of the ingress-egress node pairs in the network in reducing the rejection rates for setting up new paths.
[Chen (1999)]	Proposes a heuristic to integrate a family of routing algorithms which support applications with QoS requirements on bandwidth, delay, delay jitter, cost, path length, and their combination.
[Tomovic (2016)]	Proposes a new routing algorithm which calculates bandwidth-delay constrained routes in the fast and efficient manner.
[Yen (1971)]	Presents an algorithm for finding the K loopless paths that have the shortest lengths from one node to another node in a network.

bandwidth availability, and less throughput. Utilization representing the percentage of used capacity from a specific resource, be it node related or link metric during a specific period of time.
Delay can be distributed into several types: processing delay representing the time a node takes to process the packets headers, queuing delay, which is the time a packet spends in the routing queues, transmission delay, is the time taken by a node to push the packet bits onto the link, and propagation delay, related to the amount of time it takes for the header of the bitstream to travel from source node to the destination node [RFC7679 (2017)]. While jitter [ETSI (2014b), ETSI (2016)], representing the variation of digital signal transitions from their original positions in time. Finally, packet loss [ETSI (2016)], is the discarding of packets in a network when a router or other network device is overloaded and cannot accept additional packets at a given moment.

2.6.8.1 Network economics metrics

While implementing resource allocation, the allocating process will use the physical network resources. Therefore to assess the allocating process in terms of its consumption for the resource, cost, revenue, cost to revenue ratio, and acceptance ratio metrics are usually used. Concerning cost, it represents the number of physical resources that were used in the physical network allocating process and is given by summing up all used resources in physical nodes such as processing power, and in links such as bandwidth that have been reserved for demands. Cost is directly related to the length of the allocated physical paths, which implies that the longer the path the higher the cost. Next is revenue, which refers to the sum of the physical resources that were requested and actually allocated on the physical network, then cost to revenue ratio, which assesses and compares the allocating algorithm in terms of the costs associated due to accepting the demand, the higher the ratio the poor the efficiency of the algorithm in allocating and consuming the physical network resources.

Finally, Acceptance Ratio, which measures the overall efficiency of an allocating algorithm, and is determined by distributing the total number of demands that could be completely allocated by the physical network by the total demands received by the allocating algorithm.

2.6.8.2 General metrics

Run time of the resource allocation algorithm, which compares the resource allocation algorithms with respect to the time they require to measure the resource allocation problem altogether [Fischer (2013)].

2.7 Open research issues

Owing to the novelty of network slicing and more importantly for the wireless virtualization, there are a variety of problems that need to be resolved or at least have not been tackled optimally. In this section, we discuss the difficulties in network slicing, as well as the promises which make it interesting. The following chapters would explain some of the problems discussed in this study. However, several others are not the focus of the thesis but it is also presented here for completeness.

2.7.1 Real deployments

Just a few works have successfully evaluated their suggested slicing methods in real-world scenarios. In the wireless domain, consumers demand that actual implementations be performed to better determine the potential of a solution. Furthermore, in actual and virtual implementations, the user always changes the Base Station (BS) or AP he is connected to, but also can change the slice (if the operator or service needs to be changed). In addition, the possible strain on resource distribution and separation on a MU-AP network should be thought carefully. When deploying slices that share several BSs or APs, it may result in interference between the slices and/or unwanted load unbalance. For starters, if the spectrum were being spread cooperatively, estimates of the interference could be provided. Backtracking and solution of this dilemma should be made more precise [Zaki (2011)]. Additionally, a determination must be taken about the degree of virtualization needed to obtain a functional sliced approach for the implementation. Slicing may be performed at various speeds, but most notably at the variations between hardware and software. Virtualization can be considered at various stages with regards to networks and users such as internet providers and operators [Wang (2013)].

- Universal Virtualization, where the network is perceived as a cloud of BSs where the tenant needs to select and customize all the components to deliver the requested service, and is completely open to the operator.
- Cross-Infrastructure Virtualization, the concept of this model is to share infrastructure services among providers, and provide tenants with the opportunity to choose what resources they want.
- Minimal intra-infrastructure virtualization, which is inside a single provider's network, distribution of bandwidth only to tenants inside the network.

Deciding which strategy for network implementation is more useful remains an open issue.

2.7.2 Customer mobility plus interference

The mobility of customers is a function of wireless networks that produces a different range of issues that must be solved. Not only because mobility creates differences in links capacity and efficiency, or because it allows the amount of users on a network to vary greatly, but also because it introduces management difficulty. Wireless networks manage device accessibility which handovers, and promise QoS in view of the distance between the consumer and the connection point.

It is obvious that newly-added complications arise from giving users mobility in a network that has been sliced. Not only one user switches the BS or AP it is linked to, but also its slice . (if changing of operator or service is needed). Then, handover structures are required to cross over between slices. Since both components are operated by distinct companies and belong to individual virtual networks, it's hard to enforce. A future strategy may include a shared unified mobility manager through slices, however, this would need to be a third party agent with an open interface controller. Moreover, centralization will bring latency to the role in which deadline is quite close. Besides, a distributed approach can also be suggested. However, the delivery of mobility control has the added issue of over-sending of signaling overhead across management agencies. In summary, the issue of versatility in a simulated sliced scenario will best be addressed by the incorporation of BSs, slices and technology in order to preserve the consistency of the service.

2.7.3 Control of end users

One of the main problems of wireless resource slicing is that the users' end devices must be granted access to the wireless medium. The essence of this issue is based on the usage of wireless technologies. The most common standard in the IEEE 802.11 protocol is completely distributed, i.e., the AP does not have some power of who will transmit when an end user will transmit. The meaning of the statement is impossible to ascertain since end consumers have no influence over their computers. As opposed to 3GPP LTE, the scheduler at the base station still schedules the downlink traffic. It also has full power over the services on the uplink link. However, traffic from the endpoint is required in order to have information of how people are using the web.

2.7.4 Functions and configurations for complex wireless management

The majority of wireless equipment has complicated control functions that are vendor unique, requiring the programming of drivers and low level applications. When several slices share a base station, these basic functions must be used with caution since commands from separate slices can clash. Furthermore, each wireless connection has its own collection of configuration parameters, such as frequency of service, bit-rate, or transmit capacity, which may vary substantially from another link sharing the same infrastructure. Furthermore, in architectures with central controllers, careful attention must be provided to local functions at the machines. The probability of delays between

a central controller and physical devices means that the physical devices have a more up-to-date understanding of the local state. As a result, in some cases, physical machines will help handle their power locally. As a result, the controller would have to handle the network on a global scale, although each system will be able to make local decisions without competing with neighbouring devices.

CHAPTER 3

Methodology for Prioritized Sliced Resource Management Based on Squatting and kicking Techniques

3.1 Introduction

The Internet community has experienced an influx of new services and applications that are characterized by stringent requirements in terms of throughput, reliability, energy consumption, among others. Supporting these various services requires an agile and flexible network [ladejo (2017)]. To this effect, Network Function Virtualization (NFV) and Software Defined Network (SDN) have been envisioned as the basis for the agility and flexibility required by the future networks (e.g., 5G) [Lucena (2017)]. Service differentiation with different QoS requirements will be realized through network slices in the form of independent, mutually isolated, self-contained, logical networks consisting of both shared and reserved resources [H. Zhang (2017)]. Moreover, since the different slices are characterized by users belonging to different service groups, in principle, the different slices are attributed to different priorities. Thus, this introduces a novelty in terms of inter-slice and intra-slice prioritization. End-to-end network slicing (e.g. Access, core, Transport, Backhauls) entails slicing in both links and node resources. However, the management of link resources is a more critical part of the network slicing and presents new research challenges to be addressed (e.g., bandwidth allocation along a path, management of the prioritization on the links, and isolation between the slices in terms of traffic) compared to node resources [M. Jiang (2016), S. Xiao (2018), C. Marquez (2018), C. Marquez (2019), C. Song (2018), Sciancalepore (2017)].

In order to transport many types of services over the same network, the network must provide different QoS assurances for the different types of services, especially in congested scenarios. Service Level Agreements (SLAs) have been previously used to define the service quality experienced by traffic transiting the network and are expressed in terms of parameters such as latency, jitter, bandwidth guarantees, packet loss and downtime [Zafar (2011)]. During the past several years, many algorithms have surfaced for providing QoS for communication networks. The fundamental objective of any QoS algorithm is to ensure that excessive congestion does not occur for the packets with assured QoS. Also, it should be noted that the QoS algorithms do not create additional capacity, but only support prioritization of traffic and allocation of capacity under congested conditions, or to reduce the source rates to decrease congestion [Tata (2014), Zhang (2007)]. In today's competitive market, the service providers have rolled out revenue-generating services in their networks through assigning

applications to different classes of service and marking the traffic appropriately at the edge routers. Therefore, different services are classified into several classes [Zhang (2007)].

This multi-class and multi-priority nature of future networks makes the resource management problem non-trivial. Firstly, there exists a challenge on how to efficiently distribute the scarce network resources such as bandwidth among heterogeneous networks services characterized by a great variety of functional and non-functional requirements [Han (2018)]. Secondly, how to efficiently guarantee QoS and isolation for high priority users especially in congested scenarios while guaranteeing maximum resource utilization [M. Jiang (2016), Oliveira (2004)].

Consequently, to meet the above challenges, techniques such as network slicing will be crucial and it will require complete and effective models. These models need to be stricter on prioritization for differentiating the traffic classes under congested scenarios to improve the utilization. They also need to provide high protection for higher priority traffic class users which is crucial for the QoS guarantee [M. Jiang (2016), S. Xiao (2018), Reale (2012), Wang (2012)]. In addition, for bandwidth management, given such a multi-class scenario with prioritized demands, Bandwidth Allocation Models (BAMs) have been proposed in the past to map application requirements and priorities on a set of traffic classes. BAMs establish the amount of bandwidth per-class and any eventual resource sharing among them [Reale (2014)]. Moreover, BAMs can handle resource allocation for any resources such as bandwidth, LSPs, fiber, other [Dures (2017)]. Notably, in literature, several works treat attempt to perform dynamic bandwidth allocation for guaranteeing a given QoS level per class and optimize the utilization. These contributions are based on the Maximum Allocation Model (MAM) [RFC 4125], Russian Doll Model (RDM) [RFC 4127], Generalized RDM (G-RDM) [Adami (2007)], AllocTC-Sharing model (AllocTC) [R.F. Reale (2011)], where the main objective of these models is to guarantee a better QoS for the dynamic class of service and improve network utilization. In these models, there are different policies for bandwidth allocation for traffic demands with higher priority with respect to others [R.F. Reale (2014)]. In other words, lower priority traffic can be favored when the conditions allow it in order to make the differentiation between priorities not to be harsh. This would be based on the fact that the reserved bandwidth for high priority classes could be underutilized by the lower priority ones when applying these models. This could defeat the objective of reliable and efficient management of bandwidth that should otherwise, guarantee the QoS performance [Tata (2013)]. Nevertheless, these models need to enhance and support differentiated services together with automated, class-based, networking service provisioning.

In light of that, this chapter formally defines and evaluates, squatting and kicking techniques for self-provisioned resource sharing in multi-class networks context in order to be able to provide 100% utilization. The squatting technique enables any class of service to squat or share the unused resources from another class of service. The squatting technique allows higher priority classes of service to utilize resources reserved for lower priority ones when being unused and vice versa. For higher priority classes, it is intended to improve the utilization, increase the acceptance ratio of the demands, and guarantee no rejection of demands when there exist unutilized resources in the network exist. On the other hand, the kicking technique guarantees better QoS for higher priority traffic, where the higher priority classes can kick out lower priority ones out of their currently allocated resources. The

proposed algorithm strictly prioritizes higher priority classes in congested scenarios while operating similar to other BAMs for the non-congested scenarios.
This study has been carried out splitting the available resources in a link among the pool of classes of traffic coming from IP-Differentiated Services (DiffServ) network into the DiffServ-aware, Traffic Engineering (TE) - enabled network domain (i.e. multi-class network) according to IETF-RFC documents, to enhance the per-link total resource utilization on a class of service basis [Veres (2007), Liu (2007)]. Moreover, the proposed model can be applied to DiffServ-aware Multi-Protocol Label Switching (DS-MPLS) networks using their TE capabilities [Bahnasse (2018)].
This chapter will cover:

1. In section 3.2, it introduces the terminology that will be used along with the thesis, part of which is based on IETF-RFC documents [RFC 2474, RFC 1633, RFC 2475, RFC 2702, RFC 3272, RFC 3564]. In addition, a detailed review on alternative resource allocation models is presented.
2. In section 3.3, the chapter introduces a new policy for selecting and serving demands, which takes QoS constraints into account for different priorities/slices based on squatting and kicking techniques. Also, it introduces the mathematical definitions of SKM.
3. And section 3.4 introduces and explains the general structure of the new proposed resource allocation and QoS algorithm, including an illustrative example showing how the algorithm works, and evaluations of the algorithm.

3.2 Definitions and detailed review on alternative resource allocation models

This section has two purposes: The first one introduces a list of definitions according to IETF-RFC documents. The second purpose is to give a detailed overview about Resource Constraints Models (RCMs) such as BAMs, and Non Constrained Models (NCMs) such as First-In-First-Out (FIFO).

3.2.1 Definitions

- Demand: The number of resources required to be allocated to the network. The fundamental parameters for generating the demand are several such as source node, destination, type of resources, amount of resources requested, priority, and lifetime (period time) for an online case.
- Class-Type (CT). A CT (also class or Class of Service (CoS)): The set of traffic trunks crossing a link that is governed by a specific set of resources constraints. Where the traffic trunks are defined as an aggregate of traffic flows/demands belonging to the same class. CT is used for the purposes of resources allocation, constraint-based routing and admission control [RFC 3246].
- Preemption (P): The act of removing demand from a given path (link) in order to give room to another demand with a higher priority. Preemption is implemented by two priorities, namely, setup and holding priorities. More specifically, the preemption attributes determine whether a

demand with a certain setup preemption priority can preempt another demand with a lower holding preemption priority from a given path when there is a competition for available resources. The preempted demand may then be rerouted [Oliveira (2004), RFC 2702, RFC 2705].

- Setup priority (s): The priority of the new demand with respect to taking resources from the path (link). The setup priority is used in deciding whether this demand can preempt another demand. For preemption to occur, the setup priority of the new demand must be higher than the holding priority of the existing demand. Also, the act of preempting the existing demand must produce sufficient resources to support the new demand. That is, preemption occurs only if the new demand can be set up successfully [RFC 3209].
- Holding priority (h): The priority of the established demand with respect to holding resources in the path (link). In other words, holding preemption priority is the priority value used to determine the degree to which an active demand can maintain its assigned resources initially. When the holding priority is high, the existing demand is less likely to give up its reservation, and hence it is unlikely that the demand can be preempted [RFC 3209].
- Traffic Class (TC): The pair of class-type and preemption priority allowed for that class type. Which means that the given demand from that CT can use that preemption priority as the setup priority (s = p), the holding priority (h = p), or both (s = h = p) [RFC 4127]. TC populate the so-called multi-class networks. A multi-class network is used to transmit multiple classes of service at the same time. Therefore, the multi-class network implements the necessary mechanisms to allow specific traffic management per class.
- Reserved (CT_b, h): The total amount of the resources reserved by all the established demands that belong to CT_b and have a holding priority of h [RFC 4127].

  In this article, we define the two main strategies to handle resources (e.g., bandwidth, LSPs, fiber, slots) among classes; the Squatting and the Kicking:
- Squatting: Act or action of occupying resources allocated to other classes when their holders are not using them. It must be noted that squatting can be applied over resources allocated to either higher priority classes (default behaviour) or lower priority ones. This concept is further elaborated in the following sections [Hesselbach (2016), El-mekkawi (2018)].
- Kicking: Act or action of expelling a lower priority class from its allocated resources, either partially or totally. In the context of this chapter, we use kicking to imply the ability to remove resources from a lower priority class including both borrowed and those that are reserved for that class. Preemption, on the other hand, denotes expulsion of a lower priority class demand from resources it borrowed from other classes and not its reserved resources [Hesselbach (2016), El-mekkawi (2018)].

Any class can adopt either a squatting or a kicking behaviour. Moreover, any class can have a subject or target role in a squatting or kicking process, depending on whether it is executing the process (subject role) or it is receiving the action (target role).

3.2.2 Resource constraints models

One of the techniques that may be used to define rules and limits for link utilization for flow aggregates TCs is the BAM in IETF literature [Zafar (2011)]. BAM defines the rules that result in granting, blocking or preemption of a flow on a particular link. These models are associated and depend on the path selection algorithm (Open Shortest Path First (OSPF), Breadth-First Search (BFS), other) which defines the links in a path used by all flows. An adequate choice of the bandwidth allocation model can directly lead to improved performance of the network as a whole as well as in meeting QoS requirements defined by the SLAs. There are alternative bandwidth allocation models such as MAM, RDM and AllocTC that will be shortly described next. The above three models are based on the requirements to support DS-MPLS-TE, as described in [RFC 3564]. For the sake of keeping compatibility with RFCs, from 4125 to 4128 [RFC 4125, RFC 4127, GeraldAsh (2005), RFC 4128], and according to traffic engineering terms, the Bandwidth Constraint for class c can be defined as ($BC_c = RC_c$). Thus, the BC_c for a given class c corresponds to the initially reserved (bandwidth) resource for this class. It must be noted that, as commented in [RFC 4125], the shares for each class are not isolated. Consequently, the existence of the cross-allocated bandwidth resource cannot be obviated.

3.2.2.1 Maximum Allocation Model (MAM)

MAM is described in [RFC 4125]. It presents a simple model that allows each class of service to have a reserved bandwidth and a full share of the overall resources as far as shown in Fig. 3.1. MAM

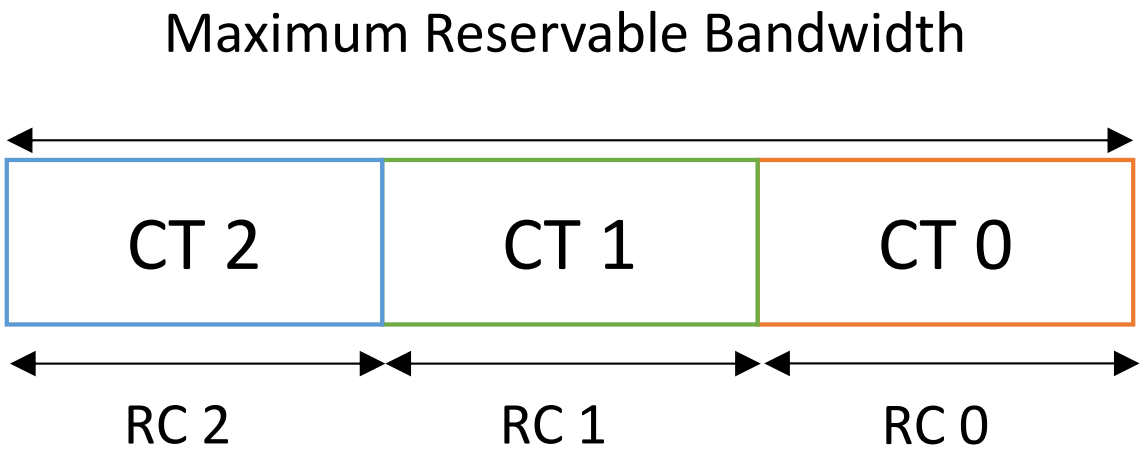


Figure 3.1: MAM allocation model

model can be described as follows:

- The sum of reserved bandwidths for all classes (considering a fixed maximum number of classes of eight) is less or equal to the maximum allocable bandwidth (less or equal to R). In general, RC_s may not be the same as the R.
- For each TC where S_i is the resources allocated for TC has $c \in [0, N-1]$ where c is the number of active class (c).

- All the active CT classes share the available bandwidth. Each CT_c can reserve a specific bandwidth quantity up to S_c. Note that S_c cannot exceed RC_c given by Eq. (3.1).
- With the restrictions, the total bandwidth allocated by all classes may not exceed the R. In this way, the sum of the total allocated resources occupied by demands S_s of a particular TC should always be less than or equal to the RC associated with this TC for a particular link given by Eq. (3.2).
- The sum of RC_s for all classes is less or equal to R. However, the sum of RC_c for $c \in [0, N-1]$ can go beyond the threshold R given by Eq. (3.3). Moreover, the sum of resource allocations of TC always corresponds to the resources available for allocation on link considered with a constraint:

$$S_c \leq RC_c \leq R \tag{3.1}$$

$$\sum_{c=0}^{N-1} S_c \leq R \tag{3.2}$$

Finally

$$\sum_{c=0}^{N-1} RC_c \geq R \tag{3.3}$$

MAM is attractive in some DS-TE environments for its simplicity and intuitiveness, easy bandwidth control policy definition, easy CoS isolation, and high resource (bandwidth) efficiency. MAM is a strict allocation model of resources. Each class has its proposed resources, and if the latter is not used, it cannot be allocated to another class. Advantage of MAM is the ability to guarantee the resources for every class within the range of resource constraints. The drawback of this model is low utilization because any class that needs more resources than itself cannot use the unused bandwidth from other classes.

3.2.2.2 Russian Doll Model (RDM)

RDM is described in [RFC 4127]. It presents a more sophisticated technique for bandwidth resource sharing among classes than MAM as shown in Fig. 3.2. RDM mechanism defines a Call Admission Control (CAC) function that blocks any new class allocation if violating a simple rule:

1. Maximum number of RC_s is equal to maximum number of CT_s = 8;
2. All demands from CT_c must use no more than RC_b (with b $\leq$ c $\leq$ 7, and $RC_b \leq RC_b - 1$, for b=1,. . . ,7), i.e.,:
3. All demands from CT_7 use no more than BC_7.
4. All demands from CT_6 and CT_7 use no more than BC6.
5. All demands from CT_5, CT_6 and CT_7 use no more than BC_5 etc.
6. All demands from CT_0... CT_7 use no more than BC_0 = R.

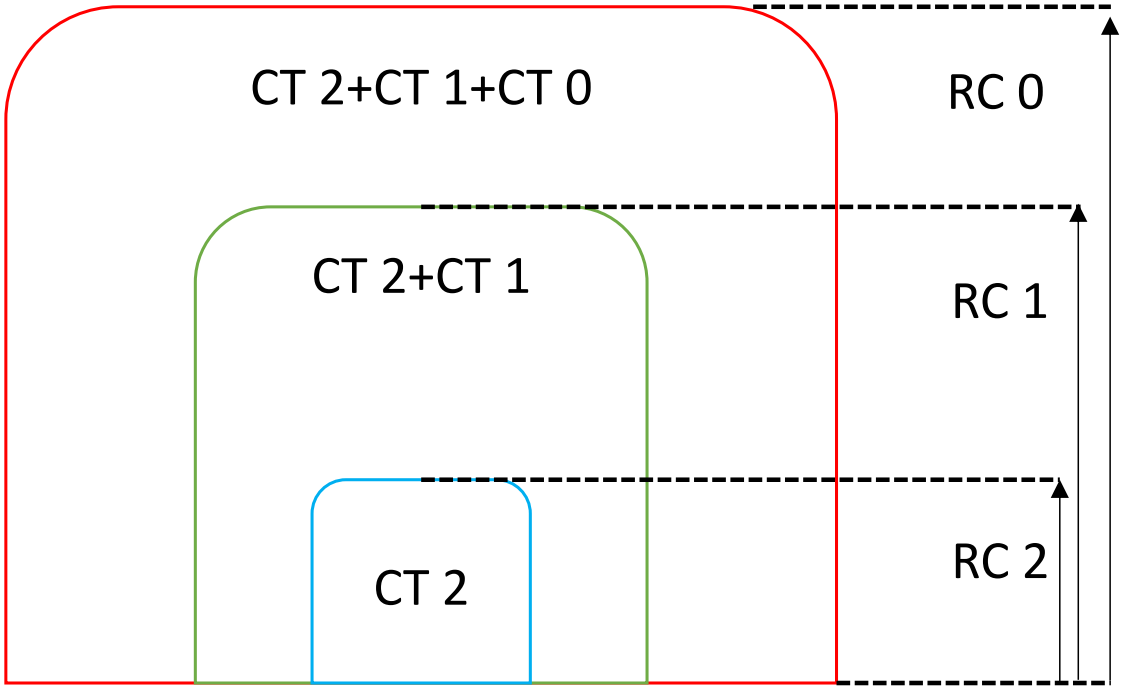


Figure 3.2: RDM allocation model

7. TC_i = (CT_c, P) where 0 ≤ i ≤ 7, 0 ≤ c ≤ 7, 0 ≤ P ≤ 7.

To illustrate the model, assume only three CTs are activated in a link and the following RCs are configured: RC0 = 160 unit, RC1 = 120 unit, and RC2 = 60 unit. Fig. 3.2 shows the model in a pictorial manner (nesting dolls). CT0 could be representing the best-effort traffic, while CT1 the non-real-time traffic, and CT2 the real-time traffic. Following the model, CT0 could use up to 100% of the link capacity given that no or traffic would be present in that link. Once it comes into play, CT1 would be able to occupy up to 75% of the link, and CT0 would be reduced to 25%. Whenever traffic would also be routed in that link, CT2 would then be able to use up to 37.5% by itself, CT1 would be able to use up to 37.5% by itself, while CT0 could use up to 25% alone.

Contrary to MAM, RDM is different by the fact that the sum of bandwidth that can be reserved by active CT_j classes where, $j \in [0, c-1]$, cannot exceed the value of the resource constraints RC_i of the CT_i. CT_i is the range of the smallest active class. In other words, i corresponds to the number of the lowest priority class Eq. (3.4). Otherwise, this upper bound RC_i which cannot be exceeded, is delimited by R. The other conditions are the same as MAM.

RDM is defined as follows:

1. For each $i \in [0, N-1]$

$$\sum_{j=i}^{c-1} S_j \leq RC_i \leq R \tag{3.4}$$

Where N is the maximum number of classes considered in the link.
The allocated resources for each class is recursively nested in the contiguous class resources (for N=8).

2. With the constraint given by Eq. (3.5).

$$\sum_{i=0}^{N-1} S_i \leq R \tag{3.5}$$

3. The Unreserved Resources (UR) information for TC_i is used by the routers, checking against the RDM parameters, to decide whether to preempt a demand. In other words, to know the exact bandwidth of any established demand from all of the resource constraints relevant to the CT associated with that demand as in Eq. (3.6).

$$\begin{aligned}(UR)_i = min[&RC_c - \sum S(CT_b, h)\ for\ h \leq P\ and\ c \leq b \leq 7,\\ &RC_{(c-1)} - \sum S(CT_b, h)\ for\ h \leq P\ and\ c \leq b \leq 7,\\ &\dots,\\ &RC_0 - \sum S(CT_b, h)\ for\ h \leq P\ and\ c \leq b \leq 7]\end{aligned} \tag{3.6}$$

Note: as the consideration of admission control rule in IETF-RFC documents, there may be more than one TC using the same CT, as long as each TC uses a different preemption priority. Also, there may be more than one TC with the same preemption priority, provided that each TC uses a different CT. The network administrator may define the TC in order to support preemption across CTs, to avoid preemption within a certain CT, or to avoid preemption completely, when so desired.
Note: according to the standard of the RFC 4127 [RFC 4127] and all other RFC documents, they assumed that the range of the preemption priority from 0 to 7, and the highest setup priority is 0 (lowest numerical value) and the lowest setup priority is 7. To prevent the preemption, the setup preemption priority should be less or equal the holding preemption priority.
In general, RDM leads to improved link utilization and optimization when compared with the MAM model. However, the general problem of the algorithms based on RDM is that the resources reservation is carried out from the bottom to top; the low priority traffic shares its resources with the higher priority traffic and not the inverse. This way the bandwidth utilization is more effective, but there are no guaranteed resources for higher priority classes.

3.2.2.3 AllocTC-sharing Model (AllocTC)

AllocTC is described in [R.F. Reale (2011)]. The AllocTC keeps RDM resource allocation strategy of Low-To-High (LTH) loans and adds the possibility of High-To-Low (HTL) loans as shown in Fig. 3.3. As such, AllocTC allows high priority classes to get resources normally reserved for low priority classes. In brief, loans are allowed in both directions (HTL and LTH). This model targets networks in which link utilization is expected to be maximized with weak isolation among TCs being acceptable. This corresponds, typically, to networks with high priority elastic applications like

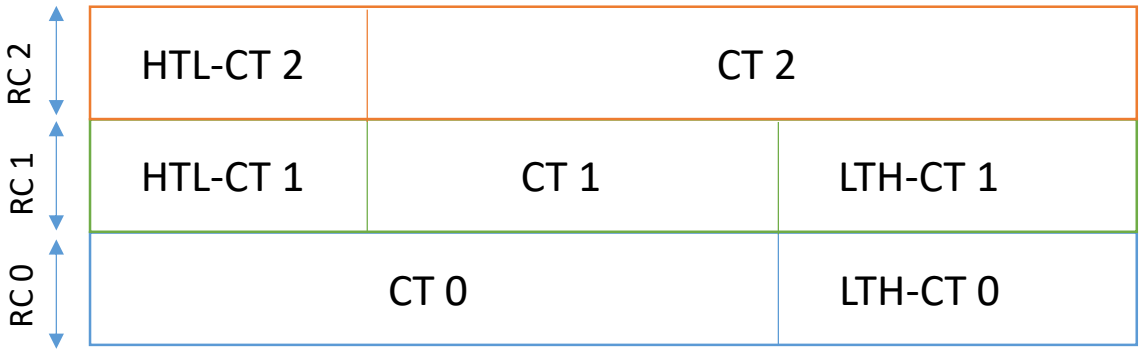


Figure 3.3: AllocTC-sharing allocation model

multimedia services, among others. AllocTC is defined as follows:

- Loan [1] "HTL" in this configurable allocation method, is the bandwidth allocated to lower priority CTs that are not being currently used may be borrowed by higher priority CTs; and
- Share "LTH" in this configurable allocation method, is the bandwidth allocated to higher priority CTs that are not being currently used may be borrowed for lower priority CTs (RDM style).

Where, S_i is the total bandwidth allocated to demands belonging to traffic class$_i$. Therefore, the maximum value for S_i can defined by Eq. (3.7) and Eq. (3.8). For each defined TC_i, a maximum allowed share (HTL_i) and (LTH_i) is defined. The HTL_i and LTH_i values should not exceed the configured RC_i.

$$HTL_i \leq RC_i \; e \; LTH_i \leq RC_i \tag{3.7}$$

$$MAX(S_i) \leq RC_i + \sum_{j=i+1}^{N-1} LTH_j + \sum_{k=0}^{i-1} HTL_k \tag{3.8}$$

AllocTC has as its main disadvantage the need to return borrowed bandwidth (in both senses). Since high-priority TCs may use bandwidth borrowed from low priority TCs, the high-priority application may be preempted.

3.2.3 Non constrained models

FIFO model is described in [McKeown (1999)]. FIFO is a method for organizing and manipulating a data buffer, where the oldest (first) entry, is processed first. It is analogous to processing a queue with first-come, first served (FCFS) behaviour: where the demands leave the queue in the order in which they arrive as shown in Fig. 3.4. FIFO is an approach for handling the demands so that the

[1] The words "Loan" and "Share" are used interchangeably.

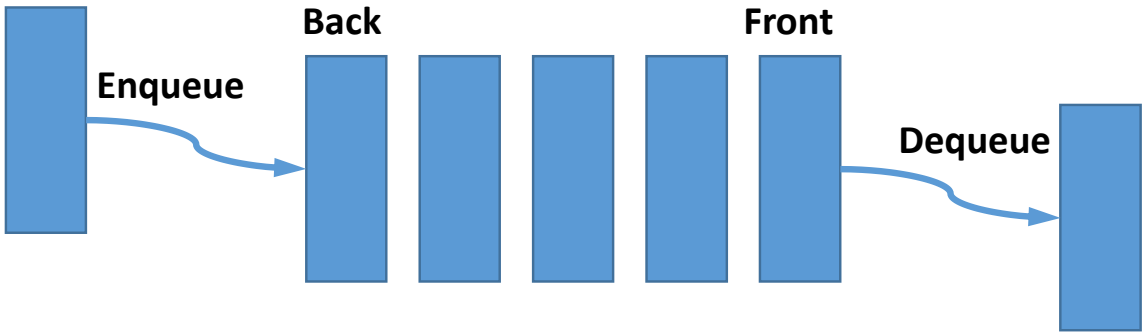


Figure 3.4: FIFO model

oldest demand is handled next. The advantage of this model that is easy to be implemented, and any demand can share resources from available resources in the network with no constraints on the links of the network. The drawback of this model is that the CoS is not considered on the link so, no guarantee for QoS.

3.3 Squatting and Kicking Model (SKM) Proposal

The need for network slicing and network virtualization for 5G networks requires models that support fast and dynamic discovery and reservation of network resources that will often be heterogeneous in type, implementation and independently administered. Thus, the main idea of our proposed SKM exploits resources partition and reservation according to different priority classes with the flexibility of using the full amount of resources when they are not demanded by other class types. This strategy is oriented to allocate the demands efficiently, but can also be used as an admission control function.

3.3.1 Assumptions

The goal of the auto-provisioning, SKM model is to achieve the more efficient dynamic allocation of the resources; motivated by the observation of the usage of the link resources, from a per-class resource usage perspective. Thus, in this work, we assumed that every single link could support up to R resources in the network where the size of R can be discrete or continuous. N is the number of classes defined in the link, and R is partitioned in classes, where RC_c is the maximum reservable resources in class c as shown in Fig. 3.5.

3.3.2 The formal specification of SKM

The overall operation results in a resource (bandwidth) allocation model that uses MAM, RDM, AllocTC integrated in a configurable way through squatting and kicking strategies to handle resources between classes/applications in a single model. Beyond that, SKM still allows new intermediate configuration settings between existing models, in this specific context of resource allocation.

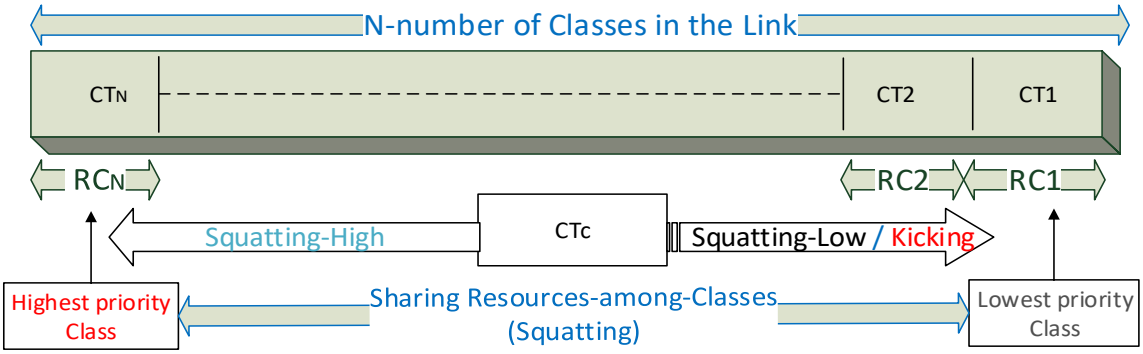


Figure 3.5: SKM-Strategy

For each demand, SKM starts working as a normal MAM algorithm (Step 1). If resources are not enough, SKM check where resources are not used, starting with higher priority classes (Step 2). This is a big difference compared to traditional schemes. If still resources are not enough or not available, in step 3 the algorithm tries to borrow resources from lower priorities. Finally, in step 4 the algorithm turns more aggressive, expelling lower priorities when no other options are available. SKM can be described and formulated according to the steps from the Alg 1 as follows:

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

$$S_c \leq RC_c \tag{3.9}$$

$$\sum_{c=1}^{N} RC_c = R \tag{3.10}$$

Eq. (3.9) ensures that the resources needed to serve the already existing demands plus the new demand do not exceed class resources constraint while Eq. (3.10), ensures that the total amount of classes resources constraints should equal to R. If constraints are satisfied, $d_j(CT_c)$ is accepted else, try step 2.

Step 2 (Squatting-High): Try to squat unused resources starting from the higher adjacent priority class upwards until there are enough resources to satisfy $d_j(CT_c)$. If there are enough resources to satisfy $d_j(CT_c)$, then accept $d_j(CT_c)$ else, try step 3. Note that the total allocatable resources in (CT_c) cannot exceed the class resource constraint RC_c plus all squatted resources from higher priority classes as in Eq. (3.11), Eq. (3.12) indicates that SH_i is less or equal to the difference between the class resource constraint and the minimum between the allocated and the reserved resources for the same class. Note that the highest priority class cannot use Squatting-High strategy.

$$S_c \leq RC_c + \sum_{i=c+1}^{N} SH_i \tag{3.11}$$

$$SH_i \leq RC_i - min(S_i, RC_i) \tag{3.12}$$

Step 3 (Squatting-Low): Try to squat unused resources starting from the lower adjacent priority

Algorithm 1 Process Assignment algorithm for SKM

Process Assignment
Loop D :Demands; Loop Demands
for *Each Demand* $d_l = d_l(CT_i) \in D$ **do**
  if $d_l \leq RC_i$ **then**
    Execute MAM Strategy
    Allocate d_l resources from the class i
  end
  else if $\exists j \;\; s.t. \;\; j > i \wedge d_l \leq RC_j - min(S_j, RC_j)$ **then**
    Execute RDM Strategy or Squatting-High
    Allocate d_l resources from CT_j
  end
  else if $\exists j$ *where* $j < i \;\; s.t. \;\; d_l \leq RC_j - min(S_j, RC_j)$ **then**
    Execute Squatting-Low Strategy
    Allocate d_l resources from (CT_j)
  end
  else
    found-kick=false
  end
  for *j=1 to i-1* **do**
    if ¬*(found-kick) and* ($\exists d_m(CT_n) \in (CT_j)$ *, and ,* $n<i$*)* **then**
      kick $d_m(CT_n)$ from (CT_j)
    end
    found-kick=true
    if ¬*(found-kick)* **then**
      Reject d_l
    end
  end
end

class downwards until there are enough resources to satisfy $d_j(CT_c)$. If the squatted higher resources plus the squatted lower resources satisfy $d_j(CT_c)$, then accept $d_j(CT_c)$ else, try step 4. Eq. (3.13) indicates that the total allocatable resources in (CT_c) cannot exceed the class resource constraint plus all squatted resources in both squatting high and low. Moreover, SL_i is working like SH_i but from lower classes, as shown in Eq. (3.14). Note that the lowest priority class cannot use Squatting-Low strategy.

$$S_c \leq RC_c + \sum_{i=c+1}^{N} SH_i + \sum_{i=1}^{c-1} SL_i \tag{3.13}$$

$$SL_i \leq RC_i - min(S_i, RC_i) \tag{3.14}$$

Step 4 (Kicking): Try to kick the assigned resources partially or totally starting from the lowest priority class upwards up to the lower adjacent class until there are enough resources to satisfy $d_j(CT_c)$. If the squatted higher resources plus the squatted lower resources plus the kicked lower resources satisfy $d_j(CT_c)$, then accept $d_j(CT_c)$ and count the kicked demands as blocked demand for the same class else, $d_j(CT_c)$ will be rejected. Eq. (3.15) ensures that the total allocatable resources

cannot exceed the class resource constraint plus all squatted resources in both squatting high and low plus all kicked resources from the lower priority classes. Moreover, the total kicked resources from lower class i K_i cannot exceed the class resource constraints RC_i as Eq. (3.16). Note that the lowest priority class cannot use kicking strategy.

$$S_c \leq RC_c + \sum_{i=c+1}^{N} SH_i + \sum_{i=1}^{c-1} SL_i + \sum_{i=1}^{c-1} K_i \tag{3.15}$$

$$K_i \leq RC_i \tag{3.16}$$

Squatting model, in any of its two high and low, is a less aggressive technique than kicking but depending on the policy needed. Therefore squatting technique is generally preferred over kicking if the class requiring extra resource allocation.

3.4 Performance evaluation

In this section, a technical comparison of SKM against the state of the art algorithm, the evaluation methodology that includes performance metrics and description of the simulations scenarios are presented. Then, we present and discuss the obtained results.

3.4.1 Technical behavior and other operational characteristics

Table 3.1 shows a set of possible behaviours and operational characteristics adopted to manage network resources for an example scenario. In other words, it is demonstrating the expected utilization and acceptance ratio depending on the available resources and load traffic in terms of the performance of SKM and for other comparative models. As example scenario of SKM, in the behavioral characteristics, SKM provides efficient resource utilization in lower priority classes only before saturation case. Also, SKM provides superior performance in the utilization of higher priority classes and the total link after saturation case. In general, SKM gives low isolation between the traffic classes due to kicking strategy. In terms of operational characteristics, SKM can share resources in both lower and higher priority classes and also SKM can kick all lower priority classes resources either the borrowed or those that are reserved for that class.

3.4.1.1 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 class (AR_c), the blocking probability per class (Bp_c) and the utilization per class (U_c) can be evaluated in Eq. (3.17-3.22) as below:

$$AR = AD/D \tag{3.17}$$

$$AR_c = AD_c/D_c \tag{3.18}$$

$$Bp = BD/D \tag{3.19}$$

Table 3.1: Technical behavior and operational characteristics comparison matrix

Behavioral characteristics	**FIFO**	**MAM**	**RDM**	**AllocTC**	**SKM**
Efficient resource utilization with high traffic load of lower priority classes	From any available resources, classes not considered	Low	High	High	High
Efficient resource utilization with high traffic load of higher priority classes	From any available resources, classes not considered	Low	Low	High	Very High
Resource utilization along the link	high	Low	Low (but better than MAM)	High	High
Accepted demands of higher priority classes along with the link	Low	Low	Low	Low	Very High
Traffic classes isolation	Not considered	High	Medium	Low	Low
Operational characteristics	**FIFO**	**MAM**	**RDM**	**AllocTC**	**SKM**
P_{HTL}	Not considered	No	Yes	Yes	Yes
P_{LTH}	Not considered	No	No	Yes	No
K_i	Not considered	No	No	No	Yes

$$Bp_c = BD_c/D_c \tag{3.20}$$

$$U = \frac{\sum_{j=1}^{D} d_j(CT_c)\, I_{A(j)}}{R} \tag{3.21}$$

$$U_c = \frac{\sum_{j=1}^{D_c} d_j(CT_c)\, I_{A_c(j)}}{R} \tag{3.22}$$

Where $I_{A(j)}$ and $I_{A_c(j)}$ denote indicator functions that give 1 if j belongs to A(j) or $A_c(j)$, respectively, and give 0 otherwise. The set A(j) corresponds to set of accepted demands and $A_c(j)$ corresponds to accepted demands by class c.

3.4.2 Offline SKM behavior

Fig. 3.6 presents the flowchart of the general procedures of the offline SKM behaviour. This behaviour introduces a new method for allocating resources to demands and facilitates resource management and reservation. In offline mode, the numbers of demands are known in advance. Therefore, in order to simplify the computation, we arrange the demands according to their priorities and size. Which means that if two demands have the same priority, the demand with a larger size for a larger amount will be allocated first to keep the utilization high in most of the cases. This is to simplify the procedure of allocating accepted demands since this strategy will make kicking not to be necessary (i.e., kicking operation becomes unnecessary since the higher priorities are processed before). Note that in this behaviour, if a connection is closed, the associated class frees all the resources it was

using. Thus, all remaining classes have to rearrange their allocated resources in order to keep as close as possible to their native service policy.

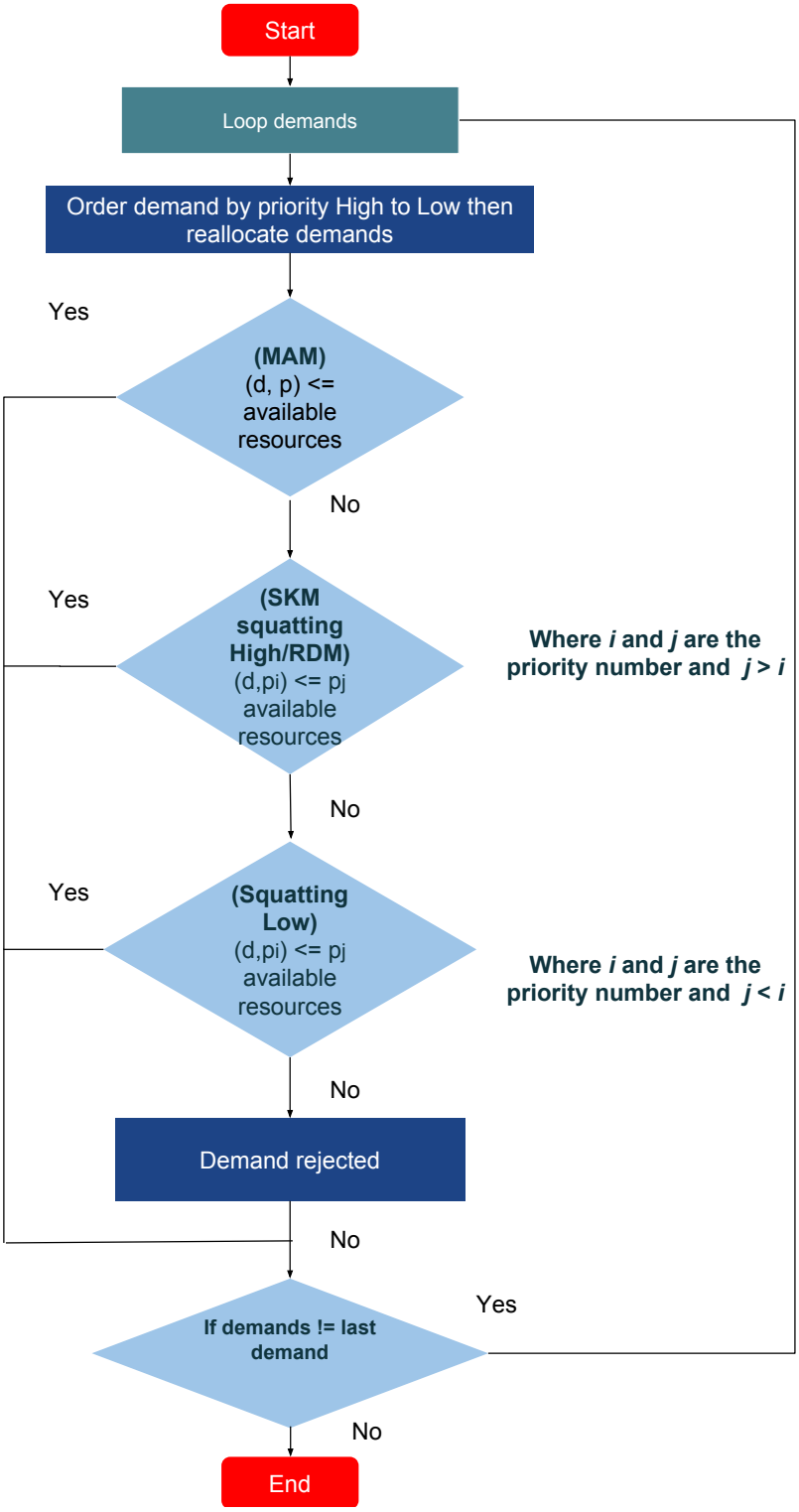


Figure 3.6: SKM Offline

3.4.2.1 Example of proposed offline SKM algorithm

Fig. 3.7 shows the network topology that consists of (2) Nodes (source to destination) and (1) link. The link in the network has a capacity equal to 40 units and divided into four priority classes. Each class has the same amount of resources equal to 10 units. Nine demands (all from source node S to destination node D) try to be mapped using available resources in the network as follows:

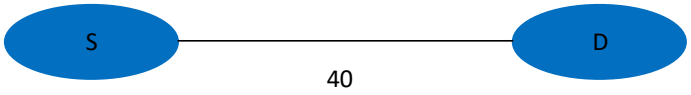


Figure 3.7: Single-link

#1: From S to D, 8 units priority 3
#2: From S to D, 4 units priority 3
#3: From S to D, 7 units priority 4
#4: From S to D, 7 units priority 4
#5: From S to D, 9 units priority 1
#6: From S to D, 6 units priority 2
#7: From S to D, 6 units priority 3
#8: From S to D, 7 units priority 2
#9: From S to D, 12 units priority 4

For an example scenario, Table 3.2 shows the SKM behaviour in the above-demonstrated example with an offline scenario in terms of allocating and reservation of resources for the demands by considering the traffic classes and the link capacities. Please note that the allocating of the demands was after the sorting process. The first allocated demand on the network is $\#9 : 12_4$, which used ten units from its priority class resources and borrowed two unused units from class 3 resources. Table 3.3 shows the results of the offline SKM algorithm in terms of the link load by each TC, U_c, U, AR_c, AR, Bp_c and Bp. From the results, class 4, accepted three demands ($\#9 : 12_4$, $\#3 : 7_4$, $\#4 : 7_4$) and class 3 accepted two demands ($\#1 : 8_3$, $\#7 : 6_3$), then the link is saturated. Moreover, all lower priority classes demands were rejected.

Table 3.2: SKM example (Offline)

# of demand : d_p 4 priority classes	**Avialable Resources**	**Allocation**
$\#9 : 12_4$	(10,10,10,10)	(10,10,8,0) SL_3
$\#3 : 7_4$	(10,10,8,0)	(10,10,1,0) SL_3
$\#4 : 7_4$	(10,10,1,0)	(10,4,0,0) SL_2
$\#1 : 8_3$	(10,4,0,0)	(6,0,0,0) SL_1
$\#7 : 6_3$	(6,0,0,0)	(0,0,0,0) SL_1
$\#2 : 4_3$	(0,0,0,0)	Rejected
$\#8 : 7_2$	(0,0,0,0)	Rejected
$\#6 : 6_2$	(0,0,0,0)	Rejected
$\#5 : 9_1$	(0,0,0,0)	Rejected

Table 3.3: SKM example (Offline) results

SKM Strategy	Class 1	Class 2	Class 3	Class 4	Link
Load by priority	10	10	10	10	40
Utilization (U)	U_1=0/40 =0%	U_2=0/40 =0	U_3=8+6/40 =35%	U_4=12+7+7/40 =65%	U=40/40 =100%
Blocking probability (Bp)	Bp_1=1/1	Bp_2=2/2	Bp_3=1/3	Bp_1=0/3	Bp=4/9
Acceptance ratio (AR)	AR_1=0/1	AR_2=0/2	AR_3=2/3	AR_4=3/3	AR=5/9

The metrics for the finite duration (online) demands considered in our work are the following:
The total acceptance ratio, AR(T): The ratio between the number of accepted demands and the total number of demands until time T. Where the observation time is from t_0 until T. Note: we assumed that once the demand is rejected, it ceases to be part of the demands in the second round or unit time (in other words leaves the system). Also, once fully served or expired, then leaves the system as in Eq. (3.23).

$$AR(T) = \sum_{\forall t \in T} \frac{(AD)_t}{D_t + (AD)_{t-1}} \times 100 \tag{3.23}$$

The total blocking probability Bp(T): The ratio between the number of blocked demands and the total number of demands until time T. The observation time is from t_0 until T Eq.(3.24).

$$Bp(T) = \sum_{\forall t \in T} \frac{(BD)_t}{D_t + (BD)_{t-1}} \times 100 \tag{3.24}$$

The acceptance ratio per class $AR_c(T)$: The ratio between the number of accepted demands by each class separately and the total number of demands for the same class until time T Eq.(3.25).

$$AR_c(T) = \sum_{\forall t \in T} \frac{(AD_c)_t}{D_{c_t} + (AD_c)_{t-1}} \times 100 \tag{3.25}$$

The blocking probability per class $Bp_c(T)$: The ratio between the number of blocked demands by each class separately and the total number of demands for the same class until time T Eq.(3.26).

$$Bp_c(T) = \sum_{\forall t \in T} \frac{(BD_c)_t}{D_{c_t} + (BD_c)_{t-1}} \times 100 \tag{3.26}$$

The utilization U(T): The ratio between the accepted or used resources in all classes within a time duration of T_j and the total capacity of resources at the time of observation Eq.(3.27).

$$U(T) = \frac{\sum_{j=1}^{D} d_j(CT_c)\, I_{A(j)}\, T_j}{R * T} \times 100 \tag{3.27}$$

Where $I_{A(j)}$ Is an indicator function equal to 1 if j belongs to A and 0 otherwise. The set A(j) corresponds to total accepted demands.
The utilization per class $U_c(T)$: The ratio between the accepted resources by each class separately within T_j and the total capacity of resources of the same class at the time of observation Eq.(3.28).

$$U_c(T) = \frac{\sum_{j=1}^{D_c} d_j(CT_c)\, I_{A_c(j)}\, T_j}{R * T} \times 100 \tag{3.28}$$

Where $I_{A_c(j)}$ Is an indicator function equal to 1 if j belongs to A_c and 0 otherwise. The set $A_c(j)$ corresponds to accepted demands by class c.

3.4.3 Online SKM behavior

Fig. 3.8 presents the flowchart of the general procedures of online SKM behaviour. By using this behaviour, the traffic of the network can be distributed fairly according to the QoS policy. This provides efficient usage of network resources and solves the online allocation problems such as the rerouting of the demands according to the priority along the unit times. In the online mode, the demands are sorted according to size and priority to minimize the number of kicking operation. The difference between the SKM behaviour in offline mode and online mode is that in the online mode the sorting is done before the process of the assignment of Alg 1 in each unit time as shown in Alg 2. Please note that either offline/online cases, sorting step improves the resource usage in the network. Because sorting according to the size tends to keep the utilization high in most cases. Moreover, sorting according to the priority guarantees the lowest amount of kicking procedure.

3.4.3.1 Example of proposed online SKM algorithm

In this example, the network topology consists of (2) Nodes and (1) link. The link in the network has a capacity equal to 40 units and divided into four priority classes; each class has the same amount of resources equal to 10 units. Also, nine demands are trying to be mapped using available resources in the network and characterized by the source node, destination node, demands size, priority and

Algorithm 2 Resource assignment algorithm for SKM Online

Procedure SKM Online D :**Load**
for *each Unit Time* t_i **do**
    $D_{selected} \leftarrow D_{(i-1)n+1:in}$ Fetch n demands sequentially from D
    $D_{checked} \leftarrow \Phi(D_{selected})$ Check Expired Demands
    $D_{sorted} \leftarrow$ SortDemands$(D_{checked})$ Sort Demands
    Process Assignment(D_{sorted})
end

duration as indicated below. Moreover, the generation rate is one demand per each unit time are as follows:

#1: From S to D, 8 units priority 3, duration 6

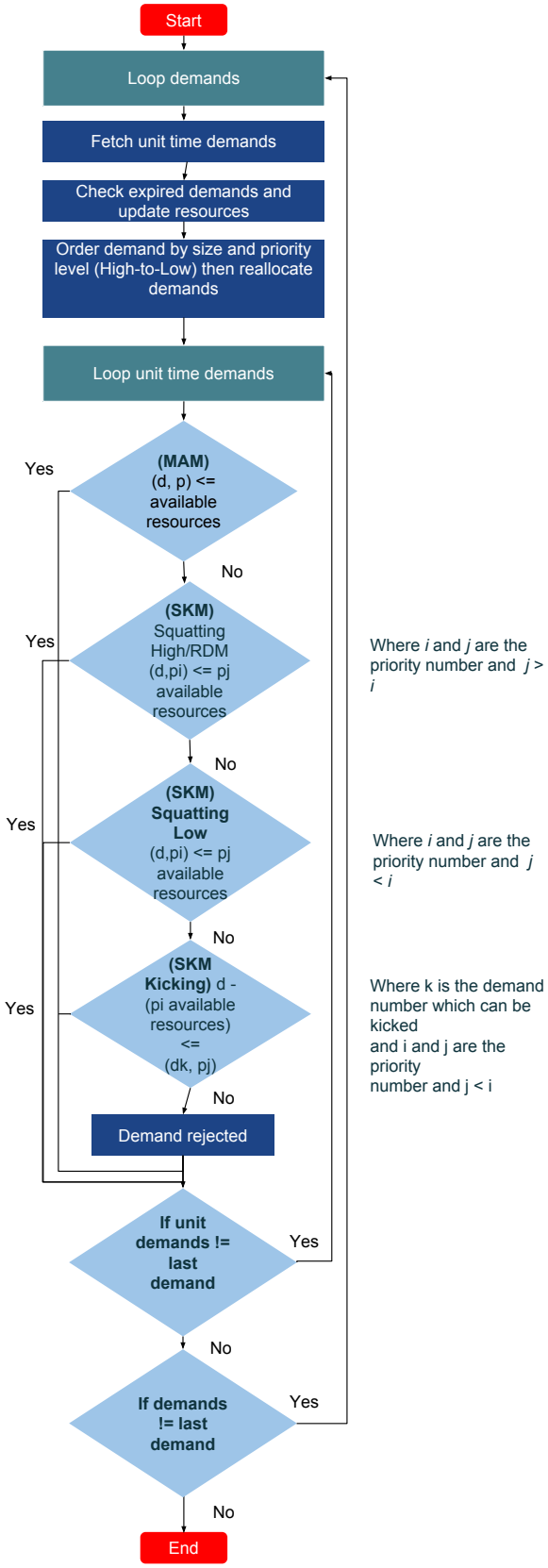


Figure 3.8: SKM Online

#2 From S to D, 4 units priority 3, duration 4
#3: From S to D, 7 units priority 4, duration 7
#4: From S to D, 7 units priority 4, duration 7
#5: From S to D, 9 units priority 1, duration 5
#6: From S to D, 6 units priority 2, duration 4
#7: From S to D, 6 units priority 3, duration 5
#8: From S to D, 7 units priority 2, duration 3
#9: From S to D, 12 units priority 4, duration 6

For an example scenario, Table 3.4 shows the SKM behaviour in the above-demonstrated example with an online scenario in terms of allocating and reservation of resources for the demands by considering the traffic classes and the link capacities. Please note that the allocating of the demands was after the sorting process in each unit. For instance, when the demand $\#3 : 7_4(7)$ arrives at the network, firstly, we must do reordering with including the new demand to the existing alive ones according to size and priority. Next, the demands$\#3 : 7_4(7)$, $\#1 : 8_3(4)$, $\#2 : 4_3(3)$ are allocated respectively. Table 3.5 shows the results of the online SKM algorithm in terms of the link load by each TC, U_c, U, AR_c, Bp_c, Bp and AR. From the results, class 4, accepted three demands until the observation time $\#9 : 12_4(1)$, $\#3 : 7_4(7)$, $\#4 : 7_4(6)$, class 3 accepted two demands $\#1 : 8_3(6)$, $\#7 : 6_3(3)$, class 2 accepted two demands $\#6 : 6_2(3)$, $\#8 : 7_2(6)$ and class 1 accepted one demand $\#5 : 9_1(3)$ then the link is saturated. Please note that low priority demands with smaller sizes, kicked $\#5 : 9_1(3)$ and $\#6 : 6_2(3)$ to satisfy the higher priority classes demands.

3.4.4 Evaluation methodology

SKM assumed distinct configurations that, intuitively, indicate that it can reproduce the behaviour of MAM, RDM and AllocTC. We complement the case study presented with a proof of concept by simulating SKM using a simple point-to-point link topology and comparing the results against the most referenced RCMs, RDM and AllocTC. Besides, we did our simulations scenarios in order to fully demonstrate the difference in the performance between the SKM and the RCMs and, also, against the most referenced NCMs, FIFO. It is important to mention that the potential flexibility and dynamic behaviour of SKM is the target of the presented simulations that is focused on validating the reproducibility characteristics of SKM model to ensure the QoS levels (especially the higher priority classes) and to achieve 100% network utilization. Moreover, five sets of simulations to evaluate the SKM performance were conducted in this chapter:

1. In the first set of simulation, we generally evaluate our proposed SKM performance in terms of link load and link load by TC as proof of concept by comparing our solution against the most referenced models, RDM, AllocTC in one scenario similar to [R.F. Reale (2011)], as explained in 3.4.5.
2. In the remaining sets of the simulations, we investigate SKM aware overall performance on limited resources networks under different traffic loads and under varying demands lifetime, in terms of link utilization, utilization per class, total acceptance and acceptance ratio per class ratio against RDM, AllocTC and FIFO, we did our scenarios in online cases as follows:

Table 3.4: Online SKM example

# of demand : $d_p(t)$	Allocation				
4 priority classes	**Unit time 1**				
	Demands expired	**New demands to be processed**	**Available resources**	**Alive demands after sorting**	**Execution**
$\#1 : 8_3(6)$	-	$\#1 : 8_3(6)$	(10,10,10,10)	-	(10,10,2,10) MAM
	Unit time 2				
$\#2 : 4_3(4)$	-	$\#1 : 8_3(5)$	(10,10,2,10)	$\#1 : 8_3(5)$ $\#2 : 4_3(4)$	(10,10,2,10) MAM (10,10,0,8) SH_4
	Unit time 3				
$\#3 : 7_4(7)$	-	$\#3 : 7_4(7)$	(10,10,0,8)	$\#3 : 7_4(7)$ $\#1 : 8_3(4)$ $\#2 : 4_3(3)$	(10,10,10,3) MAM (10,10,2,3) MAM (10,10,0,1) SH_4
	Unit time 4				
$\#4 : 7_4(7)$	-	$\#4 : 7_4(7)$	(10,10,0,1)	$\#3 : 7_4(6)$ $\#4 : 7_4(7)$ $\#1 : 8_3(3)$ $\#2 : 4_3(2)$	(10,10,10,3) MAM (10,10,6,0) SL_3 (10,8,0,0) SL_2 (10,4,0,0) SL_2
	Unit time 5				
$\#5 : 9_1(5)$	-	$\#5 : 9_1(5)$	(10,4,0,0)	$\#3 : 7_4(5)$ $\#4 : 7_4(6)$ $\#1 : 8_3(2)$ $\#2 : 4_3(1)$ $\#5 : 9_1(5)$	(10,10,10,3) MAM (10,10,6,0) SL_3 (10,8,0,0) SL_2 (10,4,0,0)SL_2 (1,4,0,0) MAM
	Unit time 6				
$\#6 : 6_2(4)$	$\#2 : 4_3(0)$	$\#6 : 6_2(4)$	(1,4,0,0)	$\#3 : 7_4(4)$ $\#4 : 7_4(5)$ $\#1 : 8_3(1)$ $\#6 : 6_2(4)$ $\#5 : 9_1(4)$	(10,10,10,3) MAM (10,10,6,0) SL_3 (10,8,0,0) SL_2 (10,2,0,0) MAM (1,2,0,0) MAM
	Unit time 7				
$\#7 : 6_3(5)$	$\#1 : 8_3(0)$	$\#7 : 6_3(5)$	(1,2,0,0)	$\#3 : 7_4(3)$ $\#4 : 7_4(4)$ $\#7 : 6_3(5)$ $\#6 : 6_2(3)$ $\#5 : 9_1(3)$	(10,10,10,3) MAM (10,10,6,0) SL_3 (10,10,0,0) MAM (10,4,0,0) MAM (1,4,0,0) MAM
	Unit time 8				
$\#8 : 7_2(3)$	-	$\#8 : 7_2(3)$	(1,4,0,0)	$\#3 : 7_4(2)$ $\#4 : 7_4(3)$ $\#7 : 6_3(4)$ $\#8 : 7_2(3)$ $\#6 : 6_2(2)$ $\#5 : 9_1(2)$	(10,10,10,3) MAM (10,10,6,0) SL_3 (10,10,0,0) MAM (10,3,0,0) MAM (7,0,0,0) SL_1 Rejected
	Unit time 9				
$\#9 : 12_4(6)$	-	$\#9 : 12_4(6)$	(7,0,0,0)	$\#9 : 12_4(6)$ $\#3 : 7_4(1)$ $\#4 : 7_4(2)$ $\#7 : 6_3(3)$ $\#8 : 7_2(2)$ $\#6 : 6_2(1)$	(10,10,8,0) SL_3 (10,10,1,0) SL_3 (10,4,0,0) SL_2 (7,0,0,0) MAM (0,0,0,0) SL_1 Rejected

- Scenario two: traffic load generated is the same for TCs of all priorities as detailed in 3.4.6.1.
- Scenario three: traffic load generated is higher for TCs of higher priority as detailed in 3.4.6.1.
- Scenario four: traffic load generated is high for TCs of lower priority as detailed in 3.4.6.1.
- scenario five: we also tested the impact of varying number demand lifetime on SKM performance as detailed in 3.4.8.1.

Table 3.5: SKM example (Online) results

SKM Strategy	Class 1	Class 2	Class 3	Class 4	Link
Load by priority (Final unit time)	10	10	10	10	40
Utilization (U)	U_1 = (9*3)/ (40*9) = 7.5%	U_2 = (6*3) + (7*2)/ (40*9) = 8.89%	U_3 = (6*3) + (8*6) + (4*4) / (40*9) = 22.778%	U_4 = (12*1) + (7*7) + (7*6) / (40*9) = 28.61%	253 / (40*9) = 67.778%
Blocking probability (Bp)	Bp_1 = 1/1	Bp_2 = 1/2	Bp_3 = 0/3	Bp_4 = 0/3	Bp = 2/9
Acceptance ratio (AR)	AR_1 = 0/1	AR_2 = 1/2	AR_3 = 3/3	AR_4 = 3/3	AR = 7/9

3.4.5 Evaluating overall performance of SKM-simulation scenario one

The overall performance of SKM was compared to RDM, AllocTC, in terms of total link load and link load by TC in a single link of MPLS network, especially under saturation case. The traffic load was generated high in the higher priority classes to evaluate the performance of each strategy before and after the saturation case. The proposed algorithm especially designed for highly congested scenarios with strict constraints for the higher priority classes. On the other hand, when the traffic is not congested the SKM behaves similar to MAM, RDM and AllocTC.

3.4.5.1 Simulation scenarios settings

The simulation described focused on the comparative validation of SKM opportunistic behaviour in respect to MAM, RDM and AllocTC.
We adopted the settings similar to [R.F. Reale (2011)] in which a single link is used as a proof of concept. The link consists of three traffic classes. The resource constraints for class 2 (highest priority class) are equal to 40% of the link capacity, resource constraints for class 1 are equal to 70% and resource constraints for class 0 are equal to 100%. The configuration parameters of the validation scenario can be summarized as follows:

- Link: 622 Mbps (STM-4 - SDH)
- Existing TC: TC0, TC1, TC2
- Table 3.6 shows the traffic classes that can be used through the bandwidth constraint of each class and obtained in the form of percentage and amount of resources.
- Number of demands equal to 1.000 and evenly distributed demand bandwidth: 05 Mbps to 20 Mbps.
- Exponential modeled demand request arrival intervals as follows: demands TC_0 - 8 s - delay of 500 s; TC_1 - 4 s - delay of 300 s; and demands TC_2 - 2 s.
- Exponentially modeled demand time life: average of 150 seconds (should cause link saturation)

Table 3.6: Bandwidth Constraints (BCs) per TCs

BC	Max BC %	Max BC (Mbps)	TC per BC
BC_0	100	622	$TC_0 + TC_1 + TC_2$
BC_1	70	435.4	$TC_1 + TC_2$
BC_2	40	248.8	TC_2

- Simulation stop criteria: number of demands

The evaluation scenario was as follows:
Traffic generated is initially higher for TCs of higher priority.
The objective of this scenario was to validate the techniques of bandwidth allocation approach of SKM and the ability to generate high admission for the higher priority classes.

3.4.5.2 Description and results evaluation

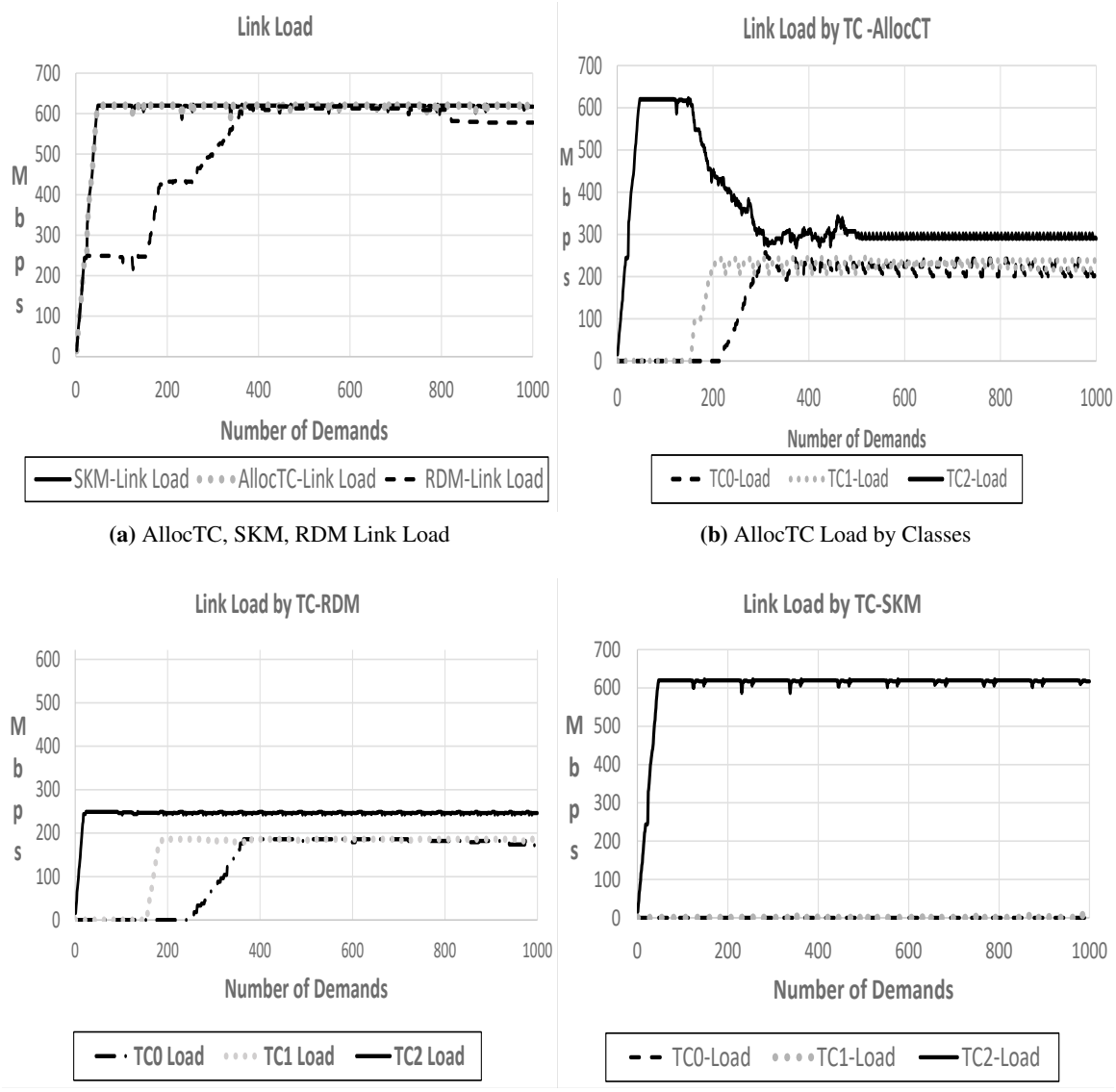
In this scenario, RDM, AllocTC and SKM are compared when highest priority TC_2 uses bandwidth above its bandwidth restriction (RC_2=BC_2) hence guaranteeing traffic competition and LTH demands in relation to TC_1 and TC_0 as shown in Fig. 3.9.
Fig. 3.9a shows that the RDM limits the link utilization to 248.8 Mbps, corresponding to BC2 configuration. This results from the fact that, in the simulation, only TC_2 demands are requested in the first 300 seconds approximately. As such, AllocTC and SKM show an improvement of link utilization in relation to RDM. Moreover, when demands belonging to TC_1 and TC_0 are requested, RDM, AllocTC and SKM reach equivalent link utilization. Unlike other models, our model accept all demands for TC_2 over the link and then TC_1 until the lowest TC_0 respectively.
The link load by TC (Fig. 3.9b and Fig. 3.9c) shows the opportunistic AllocTC behaviour with demands borrowed being returned when TC_1 and TC_0 setup required the borrowed resources. TCs load resulting from AllocTC model become similar to RDM TCs load after the borrowed resources are returned to their respective classes. Fig. 3.9d shows that in case of link saturation, the SKM gives the ability to TC_2 which is the highest priority class of traffic to kick other lower priority TCs in order to satisfy its demanded resources.

3.4.6 Evaluating overall performance of SKM-simulation scenario two, three, four

In order to evaluate our solution, the simulated topology uses one traffic source, one destination on the network consisting of a single link shown in Fig. 3.10 as proof of concept. The capacity of the link is equal to R=160 units. Moreover, the link resources divided into four classes; each class has RC_c= 40 units.

3.4.6.1 Simulation scenario settings

In these simulations scenarios, the demands are generated with a fixed lifetime of each demand equal to 1-time slot and the size of each demand is also fixed equal to 1 unit as the minimum granularity



(a) AllocTC, SKM, RDM Link Load

(b) AllocTC Load by Classes

(c) RDM Load by Classes

(d) SKM Load by Classes

Figure 3.9: Comparison of Link Load and Link Load per Class in scenario one

for allocation. Each demand has single priority generated in a static manner since we want a fixed number of demands for each priority class from (1 to 4) with a generation rate of demands per each unit time equal to 240 demand. The demands arrive at the system for service as follows: We assume that all demands for class 4 arrive first then, all demands for class 3 then, all demands for class 2 and then, all demands for class 1 in each unit time. The total number of demands among classes generated until 100 unit time is 24,000 for each scenario. Moreover, Table 3.7 shows the traffic load consideration (number of demands in each class) for validation scenarios in each class in each unit time. The main objective of the scenarios below is to analyze the performance of SKM under different loading distributions among the different priority classes. The evaluation scenarios are as

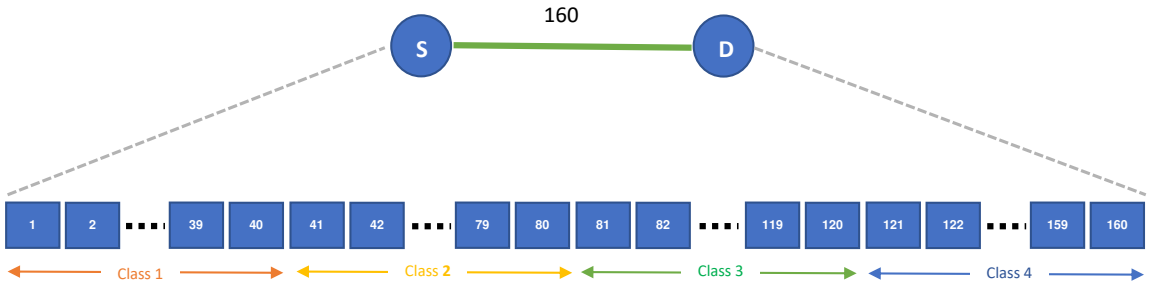


Figure 3.10: Proof-of-Concept-Simulated Topology

Table 3.7: Simulation Scenarios

Number of classes in the generating file per-each unit time	Scenario 2 Load volume Traffic (Number of demands)	Scenario 3 Load volume Traffic (Number of demands)	Scenario 4 Load volume Traffic (Number of demands)
Class-Type 1	60	20	100
Class-Type 2	60	20	100
Class-Type 3	60	100	20
Class-Type 4	60	100	20

follows:

- Scenario two: traffic load generated is same for TCs of all priorities.
- Scenario three: traffic load generated is higher for TCs of higher priority.
- Scenario four: traffic load generated is higher for TCs of lower priority.

The purpose of scenario two is to demonstrate that the SKM can guarantee to accept more demands (more strict on priorities) for higher priority classes than AllocTC and RDM in case of same loads. The purpose of scenario three is to demonstrate that SKM has an equivalent behaviour to AllocTC before the saturation case when the load is high for higher priority classes. This is verified by enforcing the share strategy of AllocTC or squatting strategy. Also, SKM achieves more accepted demands than AllocTC and RDM at high loads for higher priority classes, which is due to being stricter on priorities than the other algorithms after saturation case.
The purpose of scenario four is to demonstrate that SKM has an equivalent behaviour to RDM and AllocTC at high loads for lower priority classes. The simulation scenario enforces the share or squatting strategy that is inherent to RDM.

3.4.7 Simulation results

The performance of our proposed model is compared with RDM and AllocTC in terms of the acceptance ratio per class, utilization per class, link utilization and total acceptance ratio. The results of the simulations for all these scenarios are as shown in Figs. 3.11-3.14.

3.4.7.1 Scenario two

In this simulation scenario, Table 3.8 shows the summary of the obtained results by each model from Figs. 3.11a - 3.11f in terms of the metrics U, AR, U_c and AR_c. Table 3.8 also shows the numerical estimations (expected metric values).
The expected value of U_c for each algorithm is evaluated and presented in Eq.(3.29) as follows:

$$\mathbf{E}[U_c] = \frac{D \times p_c}{R} \tag{3.29}$$

where p_c is the probability of having a demand in class c according to the performance of each algorithm.
Please note that in this scenario the AllocTC has an equivalent performance to RDM in terms of U, AR, U_c and AR_c since in case of AllocTC, the higher priority classes borrowed unused resources from the lower ones. But when the lower classes need its resource, the borrowed resources are returned to their own classes.
The expected value of U for each algorithm is evaluated and presented in Eq.(3.30) as follows:

$$\mathbf{E}[U] = \frac{\sum_{c=1}^{N} \mathbf{E}[U_c]}{R} \tag{3.30}$$

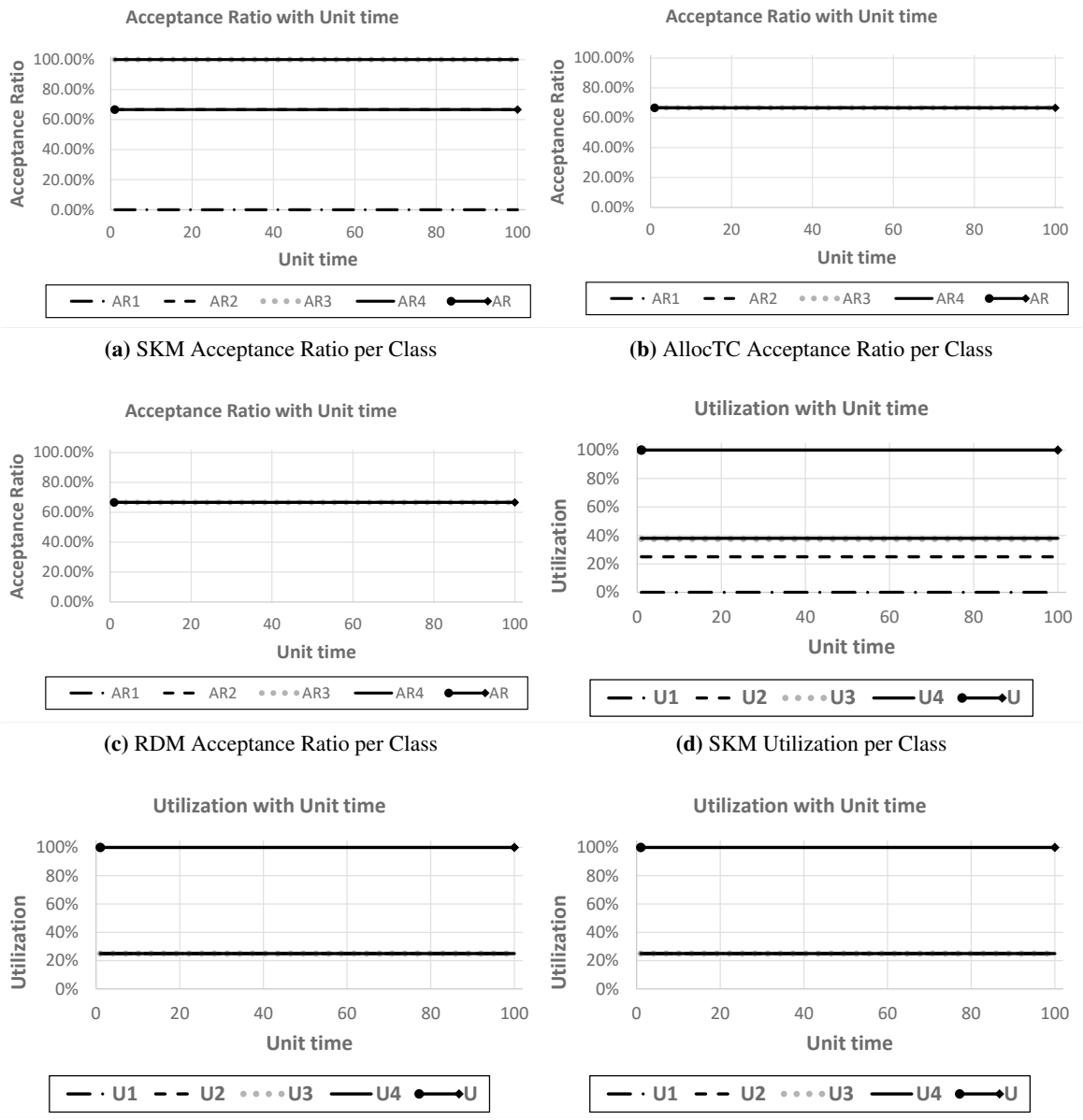
The expected value of AR_c for each algorithm is evaluated and presented in Eq.(3.31) as follows:

$$\mathbf{E}[AR_c] = \frac{D \times p_c}{D_c} \tag{3.31}$$

The expected value of AR for each algorithm is evaluated and presented in Eq.(3.32) as follows

$$\mathbf{E}[AR] = \frac{\sum_{c=1}^{N} \mathbf{E}[U_c]}{D} \tag{3.32}$$

This proves that our simulations performance gives similar results to the numerical estimations. Please note that utilization per class in terms of numerical estimations is calculated in general, from the expected performance of each class according to applied strategy. On the contrary, the acceptance ratio is calculated on specific cases depending on the number of demands in each class in each scenario. As shown in Fig. 3.11 and Table. 3.8, SKM, RDM and AllocTC resulted in 100% U and 66.67% AR where 160 demands are accepted from 240 demands per each unit time. As expected, SKM registered the highest performance among the other two strategies (RDM, AllocTC) by 33.33% in terms of AR_4. Similarly, SKM outperforms RDM and AllocTC by 33.33% in terms of AR_3 (see Figs. 3.11a - 3.11c and Table. 3.8 for models comparison in terms of AR_4 and AR_3). Further, in terms of U_c, SKM, achieved 12.5% for class 4 and, 12.5% for class 3 more than both RDM and AllocTC (see Figs. 3.11d - 3.11f and Table. 3.8).
The above results show a superior performance of SKM for class 4 and 3 in terms of both AR_c and U_c. This can be justified by the nature of SKM which permit higher priority classes to share unused resources from the lower ones and vice versa. Through the squatting technique, if there are



(a) SKM Acceptance Ratio per Class

(b) AllocTC Acceptance Ratio per Class

(c) RDM Acceptance Ratio per Class

(d) SKM Utilization per Class

(e) AllocTC Utilization per Class

(f) RDM Utilization per Class

Figure 3.11: Comparison of Utilization and Acceptance Ratio per Class in scenario two

enough resources in the link (before saturation case), the demands will be allocated with respect to the priority of the demands even if the load was high in the higher priority classes. Moreover, in the saturation case, SKM permits the higher priority users to expel the lower priority users in order to satisfy the demand requirements of the higher priority classes through kicking technique. Therefore SKM guarantees acceptance of the entire demand from class four and three as long as this demand does not exceed the available resources. The results also reveal that RDM has the same performance as AllocTC for the above classes under the considered scenario in terms of both AR_c and U_c.

Table 3.8: Summary of scenario two results

Scenario two Same load	**Simulations results**									
Metrics	U1	U2	U3	U4	U	AR1	AR2	AR3	AR4	AR
SKM	0%	25%	37.5%	37.5%	100%	0%	66.67%	100%	100%	66.67%
AllocTC	25%	25%	25%	25%	100%	66.67%	66.67%	66.67%	66.67%	66.67%
RDM	25%	25%	25%	25%	100%	66.67%	66.67%	66.67%	66.67%	66.67%
Scenario two Same load	**Numerical estimations**									
SKM	0%	25%	37.5%	37.5%	100%	0%	66.67%	100%	100%	66.67%
AllocTC	25%	25%	25%	25%	100%	66.67%	66.67%	66.67%	66.67%	66.67%
RDM	25%	25%	25%	25%	100%	66.67%	66.67%	66.67%	66.67%	66.67

This can also be justified by the nature of AllocTC which permit lower priority classes to share unused resources from the higher ones and vice versa similar to our proposal. But in case of link saturation, unlike SKM, all borrowed resources should be returned in both senses for AllocTC case. Therefore, as illustrated in this scenario settings with same traffic load in all classes, each class accepted 40 demand from 60 demands that needed to be allocated (see Table 3.7). In terms of RDM performance, the higher priority classes can not share unused resources from the lower ones so it had the same equivalent performance to AllocTC.

Table 3.9: Summary of scenario three results

Scenario three High load in lower priority classes	**Simulations results**									
Metrics	U1	U2	U3	U4	U	AR1	AR2	AR3	AR4	AR
SKM	0%	0%	37.5 %	62.5%	100%	0 %	0%	60%	100%	66.67%
AllocTC	12.5%	12.5%	25%	50%	100%	100%	100%	40%	80%	66.67%
RDM	12.5%	12.5%	25%	25%	75%	100%	100%	40%	40%	50%
Scenario three High load in lower priority classes	**Numerical estimations**									
SKM	0%	0%	37.5%	62.5%	100%	0%	0%	60%	100%	66.67%
AllocTC	12.5%	12.5%	25%	50%	100%	100%	100%	40%	80%	66.67%
RDM	12.5%	12.5%	25%	25%	75%	100%	100%	40%	40%	50%

SKM achieves the lowest performance in lower classes due to the kicking operation which results in the expelling of the lower priority users in order to satisfy the demand requirements of the high priority classes as shown in Table. 3.8, Fig. 3.11a and Fig. 3.11d. On the other hand, SKM intends to favor users belonging to high priority classes in terms of admission and resource allocation hence the observed superior performance for high classes at the expense of low priority classes. Moreover, this behaviour makes SKM a right candidate for prioritized admission control.

3.4.7.2 Scenario three

In this simulation scenario, Table. 3.9 shows the summary of the obtained results by each model in terms of utilization and acceptance ratio from Figs. 3.12a - 3.12f and compares it with the expected results.

Fig. 3.12 illustrates that the SKM outperforms RDM and AllocTC in the highest priority class by 60% and 20% in terms of AR_4 and by 37.5% and 12.5% respectively in terms of U_4 (as the expected from the behaviours). AllocTC achieves higher acceptance ratio and utilization than RDM in class 4 since in AllocTC performance the higher priority classes can borrow unused resources from the lower ones (class 4 shared 40 resources from the lower classes) as shown in Table 3.7. This is attributed to the fact that scenario three considers the higher priority classes to have more demand than the lower priority classes. Also, from the results, SKM outperforms RDM and AllocTC in class 3 by 20 % in terms of AR_3 and by 17.5% in terms of U_3 (as the expected from the behaviours) as shown in Fig. 3.12a- 3.12f and Table. 3.9. The SKM approach registers highest AR and U performance, in the higher priority classes due to the kicking operation as explained earlier. Moreover, even when the lower classes have fewer demands than the assigned resources, the unused resources can be shared by higher priority classes which is not the case with RDM. If there are any unused resources in class 1 or 2 for the case of RDM, these resources will stay idle even if there is congestion in the higher priority classes.

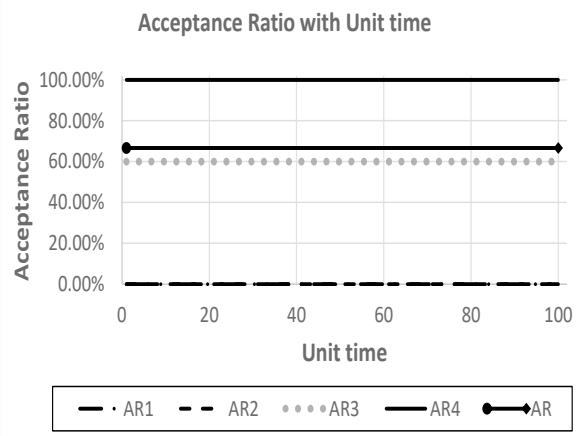
In terms of U and AR, when we increase the load in higher priority classes, the RDM performance is the lowest one among the three strategies by achieving 50% as AR and 75% as U. Where the lower priority classes can only share resources from the higher ones. So, in all unit times, the total acceptance ratio along the link cannot be 160/240 = 66.67% as in SKM and AllocTC even if the number of demands more than the capacity of the link (see Table. 3.9). This is because each class cannot exceed its resources constraints (class 1 = 20 units, class 2 = 20 units, class 3 = 100 units, class 4 = 100 units) as shown in Table 3.7.

Finally, from results of scenario three, by increasing the number of demands in the higher priority classes we can realize a significant performance difference between SKM, AllocTC and RDM approach in terms of the strictness on priority. Thus, SKM provided better performance in terms of AR and U.

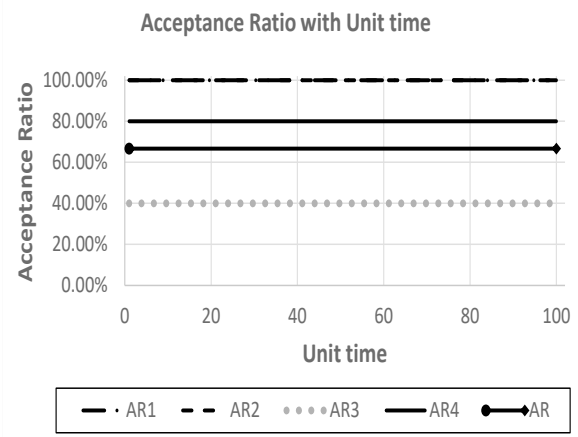
3.4.7.3 Scenario four

In this simulation scenario, Figs. 3.13a and 3.13b reflect the behavior of each algorithm in terms of U_c, U, AR_C and AR along 100 unit times.

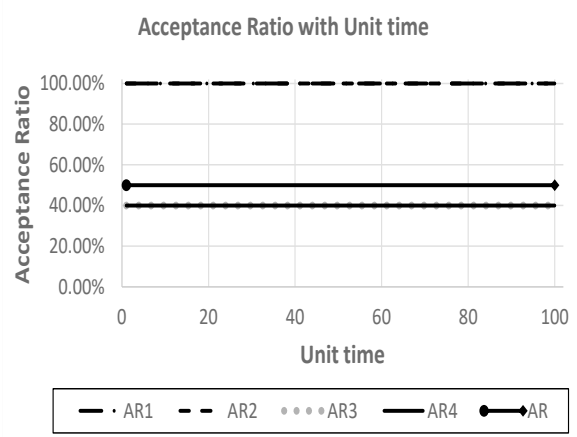
Simulations for scenario four showed that SKM, AllocTC and RDM have similar behaviour for traffic patterns in which lower priority classes have greater demands for resources. This is as expected from the performance of each algorithm since the lower classes can share all unused resources from the higher ones.



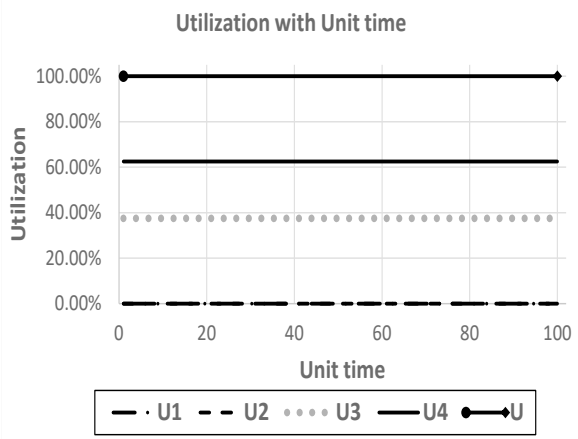
(a) SKM Acceptance Ratio per Class



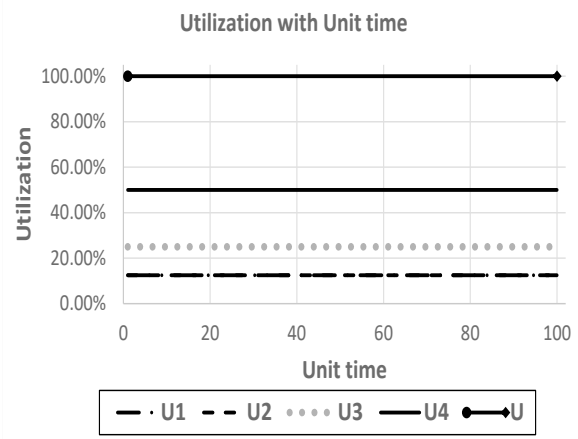
(b) AllocTC Acceptance Ratio per Class



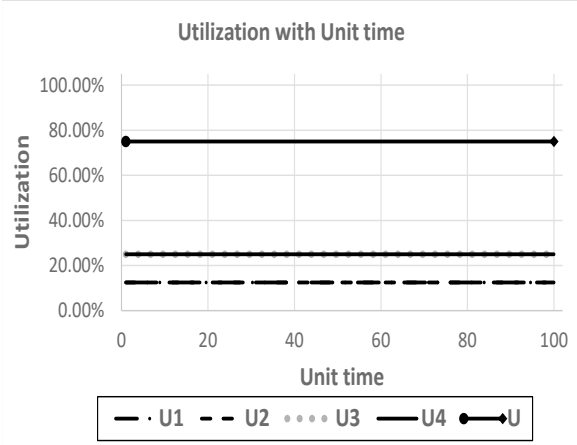
(c) RDM Acceptance Ratio per Class



(d) SKM Utilization per Class

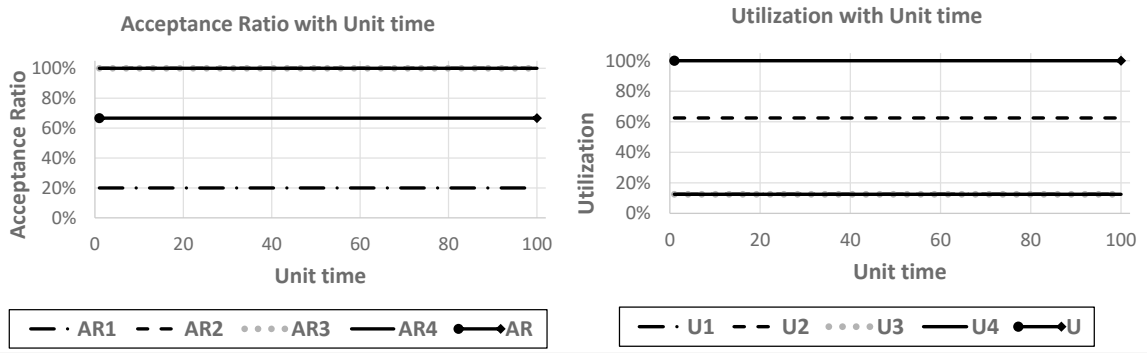


(e) AllocTC Utilization per Class



(f) RDM Utilization per Class

Figure 3.12: Comparison of Utilization and Acceptance Ratio per Class in scenario three



(a) SKM, AllocTC, RDM Acceptance Ratio per Class

(b) KM, AllocTC, RDM Utilization per Class

Figure 3.13: Comparison of Utilization and Acceptance Ratio per Class in scenario four

3.4.8 Evaluating overall performance of SKM-simulation scenario five

To evaluate the impact of the increase of demand lifetime on the performance of SKM against other state of the art algorithms, we used the same network topology of scenarios two, three and four but with varying demand lifetimes and considering a random number of demands. In this scenario, we also calculate the U, U_c, AR and AR_c. Please note that in this scenario, we compared the SKM against FIFO in order to demonstrate that our proposed model gives 100% U similar to NCMs. Besides, SKM provides a good QoS level among different priority classes.

3.4.8.1 Simulation scenario setting

In the simulations, the demands are generated with a lifetime of each demand varied randomly from 1 to 100-time slots and the size of each demand is also fixed equal to 1 unit as the minimum granularity for allocation. Each demand has a single priority generated randomly from (1 to 4) with a generation rate of demands per each unit time equal to 200 demand. The total number of demands among classes generated until 100 unit time is 20,000 demands for each simulation. Table. 3.10 shows the summary of the simulation scenario 5.

Table 3.10: Simulation Scenario five

Simulation Time:	100 Unit time
Generation Rate:	200 demands/Unit time
Capacity:	160 Units
Generation Ratio:	Class 4: 45% Class 3: 35% Class 2: 10% Class 1: 10%

The objective of this scenario is to analyze the effect of demand lifetimes on the performance of each scheme under high traffic load for higher priority classes.

3.4.8.2 Description and results evaluation

In this simulation scenario, Table 3.11 shows the summary of the obtained results by each model in terms of utilization and acceptance ratio from Figs. 3.14a - 3.14f in terms of the results from simulations for the the metrics U, U_c, AR, AR_c.

Table 3.11: Summary of scenario five results

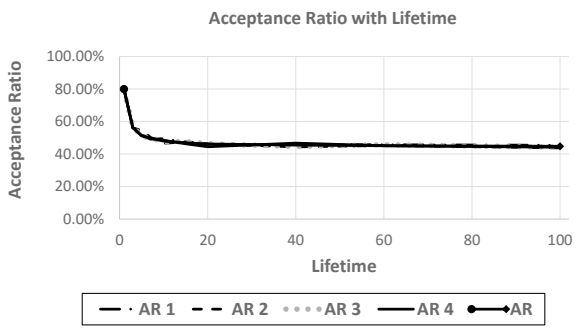
Scenario three High load in higher priority classes	**Simulations results**									
Metrics	U1	U2	U3	U4	U	AR1	AR2	AR3	AR4	AR
FIFO	10.4%	10.11%	35.42 %	44.44%	100%	51.63%	51.55%	51.37%	51.28%	51.45%
SKM	0%	0%	5%	95%	100%	0%	0%	12%	73%	51%
AllocTC	23.55%	23.55%	26.36%	26.55%	100%	76.07%	76.05%	43.69%	36.17%	51.61%
RDM	23.59%	23.55%	25%	25%	97.14%	76.11%	76.05%	40.67%	33.59%	48.43%

From Fig. 3.14 and the shown Table 3.11, FIFO (where no classes are considered), SKM and AllocTC resulted into 100% U and 51% AR as opposed to 97.14% and 48.43% achieved by RDM in terms of U and AR respectively. The reason that the RDM model offers the lowest U and AR is that in this scenario, the higher priority demands arrived at the system with a high load, but the higher priority classes cannot share resources from lower priority classes.

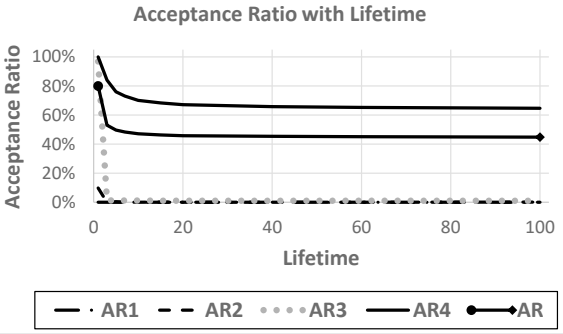
As expected, SKM outperforms RDM in class 4 by 39.41% in terms of AR_4 and by 70% in terms of U_4 (see Fig. 3.14b, Fig. 3.14d, Fig. 3.14f and Fig. 3.14h). From the results, the SKM model offers the lowest U_c and AR_c in the other classes against other schemes in this scenario as shown in Table 3.11. This is because of the increasing of the demand lifetime due to that the demanded resources to stay for a long time in the system. So, this makes it difficult to accept new demands for the other classes. Furthermore, SKM permits lower priority classes to share unused resources from the higher ones and vice versa through squatting technique even if the load was high in the higher priority classes as long as there are enough resources in the network. However, in the saturation case, SKM permits the higher priority users to expel the lower priority users in order to satisfy the demand requirements of the higher priority classes through kicking technique.

As expected, SKM outperforms AllocTC in class 4 by 36.83% in terms of AR_4 and by 68.45% in terms of U_4 (see Fig. 3.14b, Fig. 3.14c, Fig. 3.14f and Fig. 3.14g). This can be justified by the nature of AllocTC which permit lower priority classes to share unused resources from the higher ones and vice versa like our proposal, but in case of saturation, unlike SKM, all borrowed resources should be returned in both senses for AllocTC.

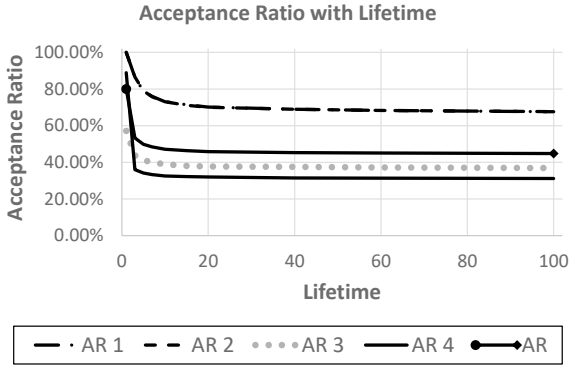
As expected, in this scenario SKM outperforms FIFO in terms of U_4 and AR_4 by 50.56% and 21.72% respectively (see Fig. 3.14b, Fig. 3.14a, Fig. 3.14f and Fig. 3.14e). FIFO does not consider classes so it cannot provide QoS. Moreover, FIFO will not result in the same utilization and acceptance ratio across the different classes due to the difference in load distributions. As expected, more loaded classes will have more acceptance and utilization since more demands from these classes will have high chances of arriving first for admission.



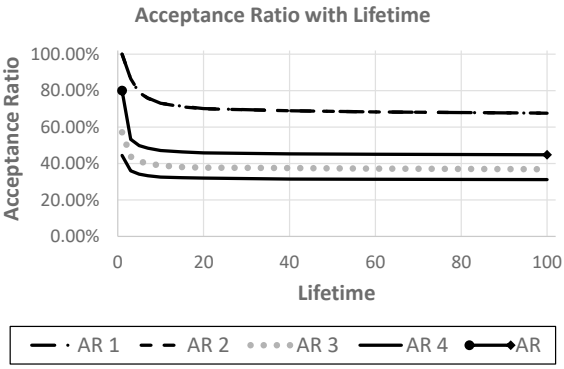
(a) FIFO Acceptance Ratio per Class



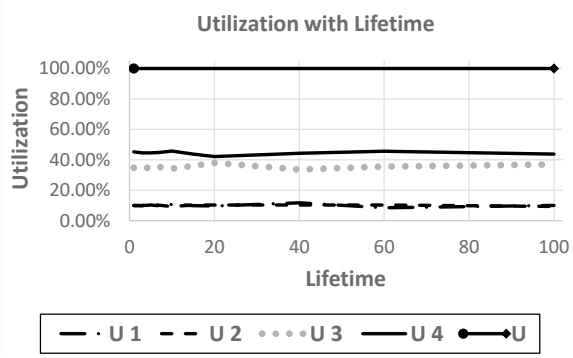
(b) SKM Acceptance Ratio per Class



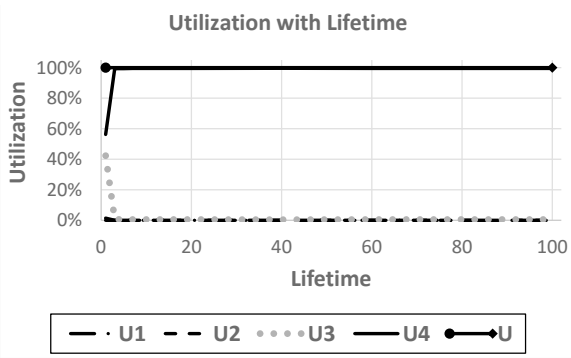
(c) AllocTC Acceptance Ratio per Class



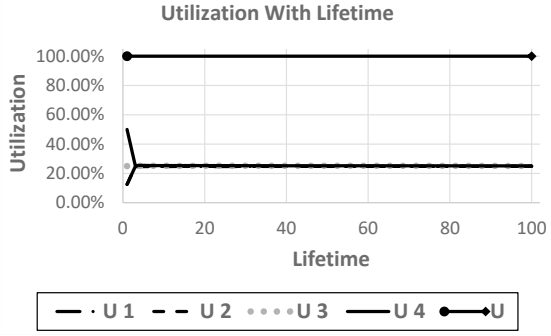
(d) RDM Acceptance Ratio per Class



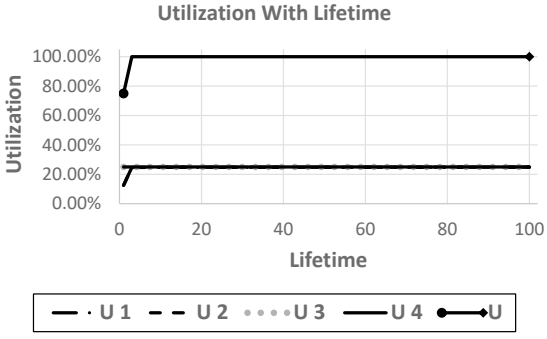
(e) FIFO Utilization per Class



(f) SKM Utilization per Class



(g) AllocTC Utilization per Class



(h) RDM Utilization per Class

Figure 3.14: Comparison of Utilization and Acceptance Ratio per Class in scenario five

These results from the considered scenario justify that SKM is better in resource management and admission control model for prioritized services than the existing sharing schemes. In other words, SKM achieves 100% as the total resources utilization (same as FIFO), and at the same time is suitable for elastic and resources eager high-priority applications. SKM is more strict on priorities by achieving and guaranteeing a good level of QoS (especially the higher ones) under large demands lifetime, which cannot be achieved by AllocTC, RDM, MAM and FIFO.

3.5 AR asymptotic value along lifetime

In this section, we will explain the behaviour of algorithms in terms of AR with growing of the demands lifetime. Table 3.12 shows the summary of the obtained results from each strategy from Fig. 3.14 in terms of AR tendency (i.e., where AR values converge).

Table 3.12: AR tendency

Strategy	AR1	AR2	AR3	AR4	AR
FIFO	44.44%	44.44%	44.44%	44.44%	44.44%
SKM	0%	0%	0%	64%	44.44%
AllocTC	66.67%	66.67%	36.36%	30.8%	44.44%
RDM	66.67%	66.67%	36.36%	30.8%	44.44%

According to RDM performance (lowest priority class can use resources up to capacity of the link, but when the higher class need its resources, preemption will be used to remove the demands from lower class) we used 160 resources as capacity of the link and for each class c, 40 as resource constraints. Moreover, we used 100 unit time (equal to 20000 demands), and the generation rate is 200 demand for each unit time as in scenario 5. Also, we set the demands from generating file to be as follows: 45% for class 4, 35% for class 3, 10% for class 2 and 10% for class 1. Please note that total number of demands equal to 20000 demand and is put randomly in the list of file. In other words, 9000/20000 = 45%, 7000/20000 = 35%, 2000/20000 = 10%. So the average number of demands per class arriving at each unit time is as follows: 90 demands for class 4 (200*45%), 70 demand for class 3, and 20 demands for class 2 and 1. From this justifications we can find out the AR_c for each class, for example with demand lifetime equal to one unit time, at start point, AR_1 = 20/20 = 100% as average and similar for AR_2 but AR_3 = 40/70 = 57.14% (because the resource constraints is equal to 40), AR_4 = 40/90 = 44.44% and AR = 120/200 = 60% as shown in Fig. 3.14d. But by increasing the demand lifetimes, the resources will be occupied for a long time in the system. As we illustrated in the previous sections, we assumed that once the demand is rejected, it ceases to be part of the demands in the second round or unit time (In other words leaves the system). Also, once fully served or expired, then it leaves the system. Thus, we can know the expected AR tendency for each class with the demand lifetimes growth (with asymptotic behaviour) by calculating the accepted demands that can be achieved by each class divided by the total arriving demands in the current unit time plus the accepted demands from the previous unit times as follows: AR_1 = AR_2= 40/(20+40) = 66.67%, AR_3 = 40/(40+70) = 36.36%, AR_4 = 40/(40+90)130 = 30.8% and AR = 160/(160+200) = 44.44% (see Table 3.12 and Fig. 3.14d).

According to AllocTC performance, either lower or higher priority classes can share unused resources from other ones and if the link is saturated, the borrowed resources will be returned in both directions. Therefore, if the demand lifetime is equal to one unit time then AR_1 = 20/20 = 100% as average and similar for AR_2 but AR_3 = 40/70 = 57.14%, AR_4 = 80/90 = 88.89% and AR = 160/200 = 80% as shown in Fig. 3.14c. Also, we can find out the expected AR tendency for each class with the demand lifetime growth as follows: AR_1 = AR_2 = 40/(20+40) = 66.67%, AR_3 = 40/(40+70) = 36.36%, AR_4 = 40/(40+90)130 = 30.8% and AR =160/(160+200)=44.44% (see Table 3.12 and Fig. 3.14c).
According to SKM performance, higher priority classes can share unused resources from the lower ones and if the link is saturated, the higher priority classes will kick the lower ones until they meet their demands. Therefore, if the demand lifetime is equal to one unit time then AR_1=0/20=0% as average and similar for AR_2 but AR_3= 70/70=100%, AR_4=90/90=100% and AR=160/200=80% as shown in Fig. 3.14b. Also, we can find out the expected AR tendency for each class with the demand lifetime growth as follows: AR_1 = AR_2= 0/(20)=0%, AR_3=0/(40+70)=0% (this is because that class 4 kicked class 3 to satisfy its resources), AR_4 = 160/(160+90)250 = 64% and AR =160/(160+200) = 44.44% (see Table 3.12 and Fig. 3.14b).
According to FIFO performance, any demand can share a resource from the available resources in the link and no classes are considered. Therefore, if the demand lifetime is equal to one unit time then AR = 160/200 = 80% on average as shown in Fig. 3.14a. Also, we can find out the expected AR tendency with the demand lifetime growth as follows: AR= 160/(160+200) = 44.44% (see Table 3.12 and Fig. 3.14a).

3.6 Summary of the findings from the simulations

RCMs are used to increase the link efficiency and admission control of users by enforcing different resource constraint for various classes of traffic so that different service QoS performance can be maximized. Therefore, it is of interest to measure the performance of RCMs by the metrics that are related to the number of Accepted/utilized demands under various operational conditions. Based on that, the performance of RDM, AllocTC, SKM and FIFO for assigned demands has been analyzed and compared. In particular, 5 different scenarios have been examined: (1) validation of our proposed model with most referenced models; (2) same load for each class; (3) increased load (number of demands) in lower priority classes; (4) increased load in higher priority classes; (5) evaluating the impact of increasing of the lifetime on the performance of SKM. We measure the QoS levels of the four strategies under these scenarios and show the trade-off between resource sharing efficiencies of the strategies.

- The simulation results showed, as in the third scenario, that the proposed model significantly optimized the link utilization, i.e., up to 100%, with strict resource constraints. Moreover the model achieved good QoS levels for the higher priority classes, i.e., 37.5%, 62.5% U_c and 60%, 100% AR_c for class3 and class 4 respectively as compared to 25%, 25% U_c for RDM and 25%, 50% U_c for AllocTC, and 40%, 40% AR_c for RDM and 40%, 80% AR_c for AllocTC as in Table. 3.9 and Fig. 3.12. The superior performance of SKM compared to the other

approaches is attributed to the fact that SKM sorts the demands according to priority and size. This is to allocate the demands from higher priority classes before the other ones to optimize the resource allocation process for the higher priority classes and improve overall network utilization. Besides, even when lower priority classes occupy resources, SKM employs the kicking mechanism to preempt the low priority users to allocate resources to the high priority classes. Furthermore, SKM permits sharing resources between lower and high priority classes, a similar per link behaviour in relation to AllocTC traffic distributions. Also, SKM modified the RDM behaviour that permits only lower priority classes to share resources from the higher ones. This can be justified by results in which RDM achieved 50% in terms of AR compared to 66.67% achieved by SKM (see Table. 3.9). This is attributed to the fact that since scenario three considered more load distribution in the high priority classes, most of the resources were used up to satisfy the demands of class 4 (highest priority class) hence little left for class3 which is lower in priority. A similar trend is observed for the case of total resource utilization where a high AR correlates to high U and vice versa.

Consequently, SKM was more strict on priorities than AllocTC and RDM under different traffic loads. This can also be justified from all scenarios results, such as the results from the second scenario, SKM achieved 100% AR_4, 100% AR_3, 66.67% AR_2, 0% AR_1 as opposed to 66.67% AR_4, 66.67% AR_3, 66.67% AR_2, 66.67% AR_1 achieved in both AllocTC and RDM (see Table. 3.8).

- In terms of AR_3 for class 3, SKM outperforms the behaviour of RDM and AllocTC in two scenarios (two, three) by realizing 100% and 60% as AR_3 compared to 66.67%, 40% and 66.67%, 40% achieved for RDM and AllocTC respectively (see Table. 3.8 and Table. 3.9).
- It should be noted that SKM gives a lower performance for class 1 and class 2 in terms of AR_c. This behaviour is expected since SKM intends to favor users belonging to high priority classes in terms of acceptance ratio, hence can be used as an approach for prioritized admission control (see Table. 3.8, Table. 3.9 and Table. 3.11).
- In terms of total resource utilization and total acceptance ratio, the simulation results indicated no significant difference in performance between FIFO, SKM and AllocTC. Furthermore, FIFO has no constraints on the link and permits resource sharing across all admitted demands without consideration of classes of services. However, for the case of FIFO, this is achieved at the expense of QoS guarantee for high priority classes. For instance, in scenario 5, SKM was observed to guarantee 95% U for the highest priority class (class 4), which is not possible by using RDM or AllocTC or FIFO. Also, SKM guaranteed 73% AR_4 compared to 33.59% for RDM, 36.17% for AllocTC and 51.45% for FIFO (see Table. 3.11).
- Regarding performance of permanent and finite duration performance demands, in the case of the permanent demand, considering scenario three (increase the load in higher priority classes), RDM registered 25%, 25%, 25% and 25% U_c across the four classes respectively, while for the finite duration demands case (lifetime equal to one) for RDM, U_c for class 1 was 12.5%, class 2 12.5%, class3 25%, class 4 25% as shown in Table. 3.9 and Fig. 3.12f.
- In case of FIFO, considering finite demands, any demand can share resource from the available

resources in the link and gives utilization in the classes from 1 to 4 as follows: 0%, 25%, 37.5%, and 62.5%, respectively similar to SKM, since we fixed the order of the generated priority demands as shown in Table. 3.9 and Fig. 3.12d. This can be justified because, in SKM performance, the demands were sorted according to size and priorities at first, to avoid using the kicking operation as a strategy to simplify the complexity of this aggressive step. After that, the process of allocation starts.

On the other hand, for FIFO, considering permanent duration demands the results of the average utilization for classes from 1 to 4 were as follows: 0%, 0%, 0%, and 100%, respectively. It is observed that in both permanent demands and finite duration demands cases, FIFO and SKM gave the same performance in terms of acceptance ratio and utilization across all the classes. This can be justified by the results in which RDM offers higher performance for lower classes either for the permanent duration demands case or for the finite demand case since the higher priority classes limit its resources.

- In case of AllocTC, considering permanent demands, the results of AR were as follows: for class 1 = 25%, for class 2 = 25%, for class 3 = 25% and class 4 = 25%. For AllocTC, considering finite duration demands, the results of AR were as follows: for class 1 = 12.5%, for class 2 = 12.5%, for class 3 = 25% and fro class 4 = 50% as shown in Table. 3.9 and Fig. 3.12e. This is attributed to the fact that when demands arrive with finite duration case, the unused resources can be allocated to the higher priority classes until the lower priority class users reclaim these resources through the preemption mechanism.
- We also analyzed the impact of processing and time costs. The proposed algorithm behaviour has a sorting step, which requires slightly more memory, but we did not measure and focus on the cost in terms of memory since our focus was the run time of the algorithms. SKM achieved 1 hour, 4 minutes, 54 seconds and 77 milliseconds as average runtime to serve the demands after running the algorithms 20 times using scenario 3. RDM and AllocTC have a slightly lower run time complexity (35 and 20 minutes respectively) than SKM. However, SKM provided very high utilization and acceptance ratio in higher priority classes, as shown in Table. 3.9 and Fig. 3.12. Also, when we compare the proposed algorithm with FIFO, SKM's run time complexity is approximately 45 minutes more than FIFO. Please note that in general, the processing cost and time infinite demands case will be more than in permanent demands case which is attributed to the fact that the sorting is done in each unit time under finite demands case, while in the case of permanent demands case the sorting operation is performed once. Please also note that the used computer had Intel(R) Core(TM) 2 CPU 6400 @ 2.13GHz Memory 6GB.

From the above results, SKM turns out to be a smart strategy for prioritized admission control compared to RDM and AllocTC in both, permanent demands and finite duration demands cases. This is because SKM can allow greater sharing of resources among different classes, and guarantee high QoS for high priority classes in all the test scenarios. Moreover, higher resource utilization efficiency is achieved due to the flexible sharing of resources in SKM. It also registers a better global resource utilization compared to RDM in both traffic scenarios and the same performance as FIFO

and AllocTC. These results justify that SKM is a better resource management and admission control model for prioritized services than the existing schemes.

3.7 Conclusions

BAMs are of great value in the context of efficient and customized use of network resources. Therefore, in this chapter, we formally defined the SKM techniques (i.e., online and offline SKM) for strict constraints and validated the SKM techniques against other states of art algorithms. Moreover, we demonstrated that SKM could provide full utilization, clearly differentiate priorities, and strictly prioritize resource allocation to higher priority classes as opposed to other proposals. The SKM starts working as a simple MAM algorithm, very conservative. However, the behaviour changes when more resources are requested and it gets more aggressive when higher priorities are not able to get enough resources. Simulations have validated the SKM considering the performance in a single link in terms of utilization and acceptance ratio, including metrics per priority class. The proof of concept and the results of our simulations showed that thanks to our proposed SKM model, we cannot only significantly optimize the overall network utilization but also achieve proper QoS levels (especially the higher priority ones). SKM was compared to the RDM and AllocTC for cases of permanent and finite duration demands. In RDM, the reservation of resources is made from bottom to top and not the reverse. So, in this way, the resource utilization is more effective in comparison to MAM, which does not permit resource sharing across classes, but in this case, there is no guaranteed bandwidth for higher priority classes. Therefore, the benefit of using SKM is that the given class can use the unused resources from other classes (high or low) by means of initiating a squatting process, this is similar to the to AllocTC in per link behaviour of traffic distribution scenario. Beyond that, in SKM, the usage of resources for the higher priority classes is greater than originally reserved. SKM guarantees 100 percent of admission of high priority demands as long as there are resources in the lower priority classes regardless of whether these resources are unused or occupied by the lower priority classes by means of initiating a kicking process. It is expected that groups of higher priority applications on multi-service networks could benefit from improved link utilization achieved by SKM. This corresponds to dynamically providing support to improve the quality of the application (SLA) for traffic distributions that occur in actual network operation, which means that the SKM is strict on priorities more than AllocTC and RDM.

As for the case of the FIFO approach, the demand can share any available resources from the link, but the problem is the demands can be coordinated or handled from oldest to newest only with non-definition of classes of service (no-constraints in the links) so, no guarantee for QoS. However, also applying SKM model, the performance is the same or very close to FIFO in terms of the scalable distribution of resources from either low or high classes and in addition to the feature of providing the quality of service by considering the priorities in the link.

From the simulations results, irrespective of the load distribution among classes, such as in scenario two, SKM was found to guarantee 100% acceptance ratio for the higher priority users (class 4, class 3) whenever the higher priority demand does not exceed the available network resources as compared

to 66.67% for RDM and AllocTC respectively. The SKM Model can reproduce the behaviour of MAM, RDM, and AllocTC in a single model and, as such, generalizes the inherent behaviour of these BAMs in a single implementation.

An important advantage of adopting SKM instead of a single BAM model instance in a network is to provide network managers with a single solution (model) that allows the optimization of network and link utilization with different load profiles. In effect, SKM provides some adaptability since it may be configured to have distinct behaviour for distinct load profiles.

Another SKM inherent advantage that has not been totally explored in this chapter is that, since it is a single model, the rules for preemption and shares may be adjusted to provide a smooth migration among the behaviour of current existing BAMs. In fact, SKM may potentially cope with the dynamics of the network traffic load profile and have sets of configured behaviours for them, including transition patterns of behaviours. Beyond that, SKM still allows new intermediate configuration settings between existing models, in this specific context of resource allocation. In effect, it is now possible with SKM to define Kicking strategy, Squatting-High and Squatting-Low for all traffic classes. New allocation strategies include the integration of the three strategies to provide a set of additional capabilities that might be capable of supporting new classes of traffic load profiles that have not been supported each of the above in a single or multi-BAM implementation.

Finally, SKM is a suitable strategy regarding some emerging technologies that are characterized by diverse QoS requirements and prioritized admission control. This is typical of 5G networks which are expected to serve flexible and diversified requirements hence the need to allocate resources dynamically.

CHAPTER 4

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

The main motivation of this chapter is to extend the work of Chapter-3 by using the basis where we proposed the best BAM method up to the present time in terms of utilization and capacity to organize demands belonging to different categories. However, in the current chapter, what we do is to take this strategy to be used in this new problem in the field of network slicing for 5G networks, considering intelligent decisions and excellent performance provided by SKM, as demonstrated in the previous chapter. 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 chapter 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 the proposed and analyzed 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 can not be achieved by other existing schemes due to priority constraints.

4.1 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 [3GPP1 (2020), 3GPP2 (2020)]: 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 [L. Feng (2020), A. Huang (2020)]. With NFV, 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 [Ma1 (2020), Ma2 (2020), Reyhanian (2020), A. M. Medhat (2017), Li (2017), Zhou (2016)].

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 [ETSI (2013a)] and suggests the OpenSource MANO (OSM) [OSM (2019)] 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.

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 as shown in Chapter-3. We, therefore, consider the SKM algorithm for the work in this chapter.

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

In light of that, in this work, we are exploiting the results from the previous work of Chapter-3 in the following aspects:

- Proposing a service deployment algorithm based on the intelligence of the SKM model defined on Chapter-3 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 Chapter-3 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 previous chapter (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 previous chapter. 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.

In summary, our contributions in this chapter are the following:

1. This chapter proposes 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 [Reale (2016)].

4.2 Network model and problem formulation

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

4.2.1 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 $\in L$ to denote a single edge substrate link between substrate node i $\in X$ and substrate node j $\in X$. Each substrate link l is characterized by i) Maximum link resources capacity R(l); ii) Available link resources at a given time denoted by $R_a^t(l)$; iii) Consumed link resources $R_z^t(l)$ at time t; iv) A set of traffic slices assigned along the link are denoted by CTs(l), where CT_N(l) is the highest priority slice and CT_1(l) is the lowest priority slice; v) Actually allocated resources to slice c S_c(l), where $c \in [1, N]$; vi) Slice resource constraints RC_c(l). 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 $P_{s,r}^k$

4.2.2 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.

4.2.3 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 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

$$Max\ \mathbf{U}(\mathbf{T}) \tag{4.1}$$

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:

$$U(T) = \frac{1}{T}\sum_{t\in T}\frac{1}{L}\sum_{\forall l\in L}\sum_{w\in W}\frac{X_w^{t,l} * d_w(CT_c)}{R(l)} \tag{4.2}$$

Where $X_w^{t,l} \in [1,0]$ is a binary variable equal to 1 if resources are allocated to the request $w \in W$ on the link $l \in L$, 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 $l \in L$ at any unit time t descried by:

$$R_z^t(l) = \sum_{w\in W} X_w^{t,l} * d_w(CT_c) \tag{4.3}$$

This goal is subject to the following:

1. link constraints:

$$\sum_{\forall l\in P_{s,r}^n} RC_c(l) \leq R(l), \forall t \tag{4.4}$$

$$Z(d_w) = R_a^t(l) \geq P_{s,r}^k * d_w(CT_c), \quad \forall t, \forall l \in P_{s,r}^k, w \in [1, D] \tag{4.5}$$

Eq. (4.4) ensures that the maximum reservable bandwidth for a link l at any time is less than or equal to the link capacity for that link. Eq. (4.5) 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 Chapter-3. 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:

$$S_c(l) \leq RC_c(l) \tag{4.6}$$

$$\sum_{c=1}^{N} RC_c(l) = R(l) \tag{4.7}$$

   Eq. (4.6) ensures that the resources needed to serve the already existing demands plus the new demand do not exceed slice resources constraint while Eq. (4.7), ensures that the total amount of slices resources constraints should be equal to $R(l)$. 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 $CT_c(l)$ cannot exceed the slice resource constraint $RC_c(l)$ plus all squatted resources from higher priority slices as in Eq. (4.8). Eq. (4.9) indicates that $SH_q(l)$ 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.

$$S_c(l) \leq RC_c(l) + \sum_{q=c+1}^{N} SH_q(l) \tag{4.8}$$

$$SH_q(l) \leq RC_q(l) - min(S_q(l), RC_q(l) \tag{4.9}$$

   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. (4.10) indicates that the total allocatable resources in $CT_c(l)$ cannot exceed the slice resource constraint plus all squatted resources in both squatting high and low.

Moreover, $SL_q(l)$ works like $SH_q(l)$, but from lower slices, as shown in Eq. (4.11). Note that the lowest priority slice cannot use Squatting-Low strategy.

$$S_c(l) \leq RC_c(l) + \sum_{q=c+1}^{N} SH_q(l) + \sum_{q=1}^{c-1} SL_q(l) \tag{4.10}$$

$$SL_q(l) \leq RC_q(l) - min(S_q(l), RC_q(l) \tag{4.11}$$

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. (4.12) 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, $K_q(l)$ cannot exceed the slice resource constraints $RC_q(l)$ as Eq. (4.13). Note that the lowest priority slice cannot use kicking strategy.

$$S_c(l) \leq RC_c(l) + \sum_{q=c+1}^{N} SH_q(l) + \sum_{q=1}^{c-1} SL_q(l) + \sum_{q=1}^{c-1} K_q(l) \tag{4.12}$$

$$K_q(l) \leq RC_q(l) \tag{4.13}$$

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 thesis, 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.

4.3 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. 4.1. This is because the traffic flowing through VNF-FGs can exhibit high throughput and high burstiness in the dynamic nature of bandwidth-intensive services.

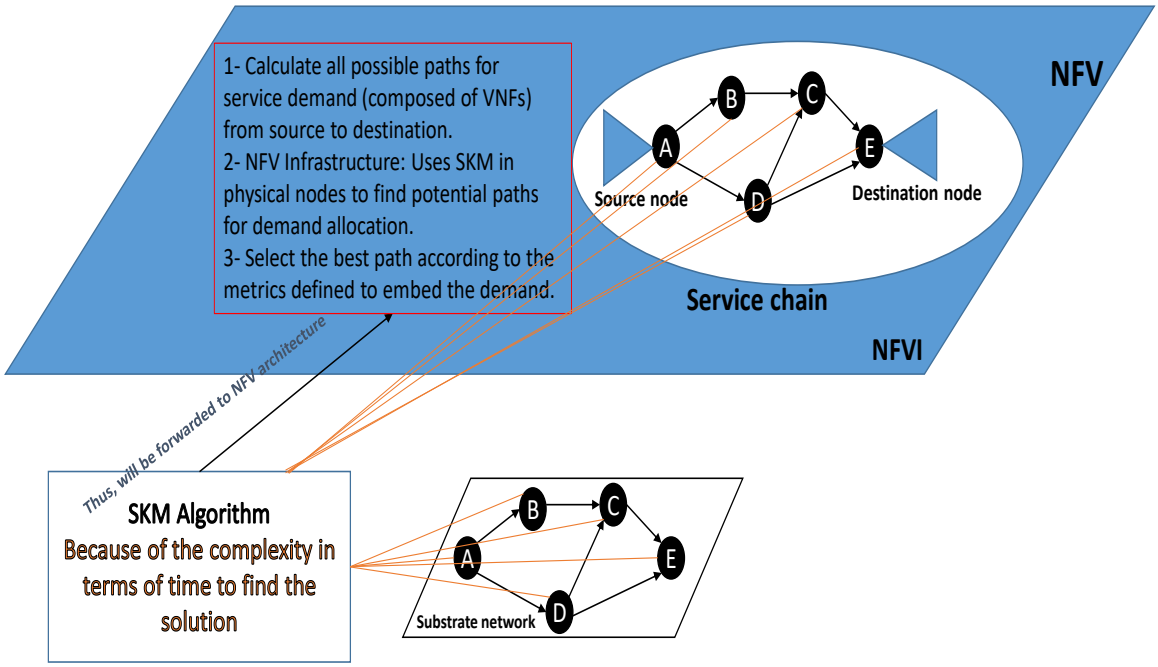


Figure 4.1: An illustrative diagram showing how SKM organizes the execution of SFC-VNFs on a shared underlying network to allocate demands

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.

4.3.1 Routing algorithm step

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

4.3.2 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.

Alg 3 summarizes the steps of SKM to allocate the demand in a multi-slice network.

4.3.3 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.
For task 1, the available link resources at any time can be calculated as below:

$$R_a^t(l) = R(l) - R_z^t(l) \tag{4.14}$$

$$R_z^t(l) = \sum_{c=1}^{N} R(l) - (RC_c(l) - min(S_c(l), RC_c(l))) \tag{4.15}$$

Eq. (4.14) shows the calculation of available resources in a link. Moreover, $R_z^t(l)$ can be determined

Algorithm 3 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* $l \in P_{s,r}^k$ **do**
    if $d_w(CT_c) \leq$ *slice c constraints* $CT_c(l)$ **then**
      Execute MAM strategy using Eq. (4.6)
    end
    else if *the total allocatable resources in slice c* $S_c(l) \leq$ *the sum of* $CT_c(l)$ *and all squatted resources from higher priority slices* **then**
      Execute RDM strategy using Eq. (4.8)
    end
    else if $S_c(l) \leq$ *the sum of* $CT_c(l)$ *and all squatted resources in both higher and lower priority slices* **then**
      Execute Squatting-Low strategy using Eq. (4.10)
    end
    else
      found-kick=false
      for *all slices priority* $<$ *slice c priority* **do**
        if $S_c(l) > CT_c(l)$ *plus all squatted resources in both squatting high and low* **then**
          Execute Kicking strategy using Eq. (4.12)
          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

by the summation of the difference between $R(l)$, and the minimum between assigned and reserved resources for each slice as in Eq. (4.15), where N is the number of slices along the link l.
Then, the link's available resources is the minimum value of $R_a^t(l) = Min(R_a^t(l))$.
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. (4.16):

$$Max\left\{Min(R_a^t(l)) \geq P_{s,r}^k d_w(CT_c)\right\}, \quad \forall l \in P_{s,r}^k, k \in [1, K] \tag{4.16}$$

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. (4.17).

$$Min\left\{\sum_{\forall l \in P_{s,r}^{k}} R_{z}^{t}(l)\right\}, k \in [1, K] \tag{4.17}$$

Alg 4 summarizes the procedure for highest available routing path selection. After defining the

Algorithm 4 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* $l \in path\ P_{s,r}^{k} \in A_p$ **do**
  Calculate available resources of a link l
  $R_z^t(l) = \sum_{c=1}^{N} R(l) - (RC_c(l) - min(S_c(l), RC_c(l)))$
  Determine path available resources
  $R_a^t(l) \Leftarrow$ Min$(R_a^t(l))$
end
for *each* $(P_{s,r}^{k} \in A_p$ *and* $R_a^t(l) > 0)$ **do**
  Select the optimal path based on highest available resources
  $Max\left\{Min(R_a^t(l)) \geq P_{s,r}^{k} d_w(CT_c)\right\}, \forall$ l $\in P_{s,r}^{k}, k \in$ [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
    $Min\left\{\sum_{\forall l \in P_{s,r}^{k}} R_z^t(l)\right\}, k \in [1, K]$
  end
end

optimal path that meets QoS constraints, the demand will be allocated based on the SKM in the network.

4.4 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.

4.4.1 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 4.1 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 4.4.4. 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.

Table 4.1: Summary of the main attributes of the comparison algorithms

Algorithm	Key attributes
MAM	I) It is a strict allocation model of the link resources. Each $CT_c(l)$ has its private resources, and if the latter is not used, it cannot be allocated to another $CT_c(l)$. 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	I) It is a nested allocation model of the link resources. The highest $CTs(l)$ priority can reuse the free resources of lower priority $CTs(l)$. 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	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	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	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 $CTs(l)$. 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.

4.4.2 Offline and online behaviors of the proposed deployment policy

The deployment policy proposed in this chapter 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.

4.4.2.1 Offline behavior of the proposed deployment policy

Fig. 4.2 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;
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).

4.4.2.2 Online behavior for the proposed service deployment policy

Fig. 4.3 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 3 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 4.3 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 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.

4.4.3 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 [Chowdhury (2012)], [Kibalya (2020)]. All the performance metrics used are described below for each one of the behaviors.

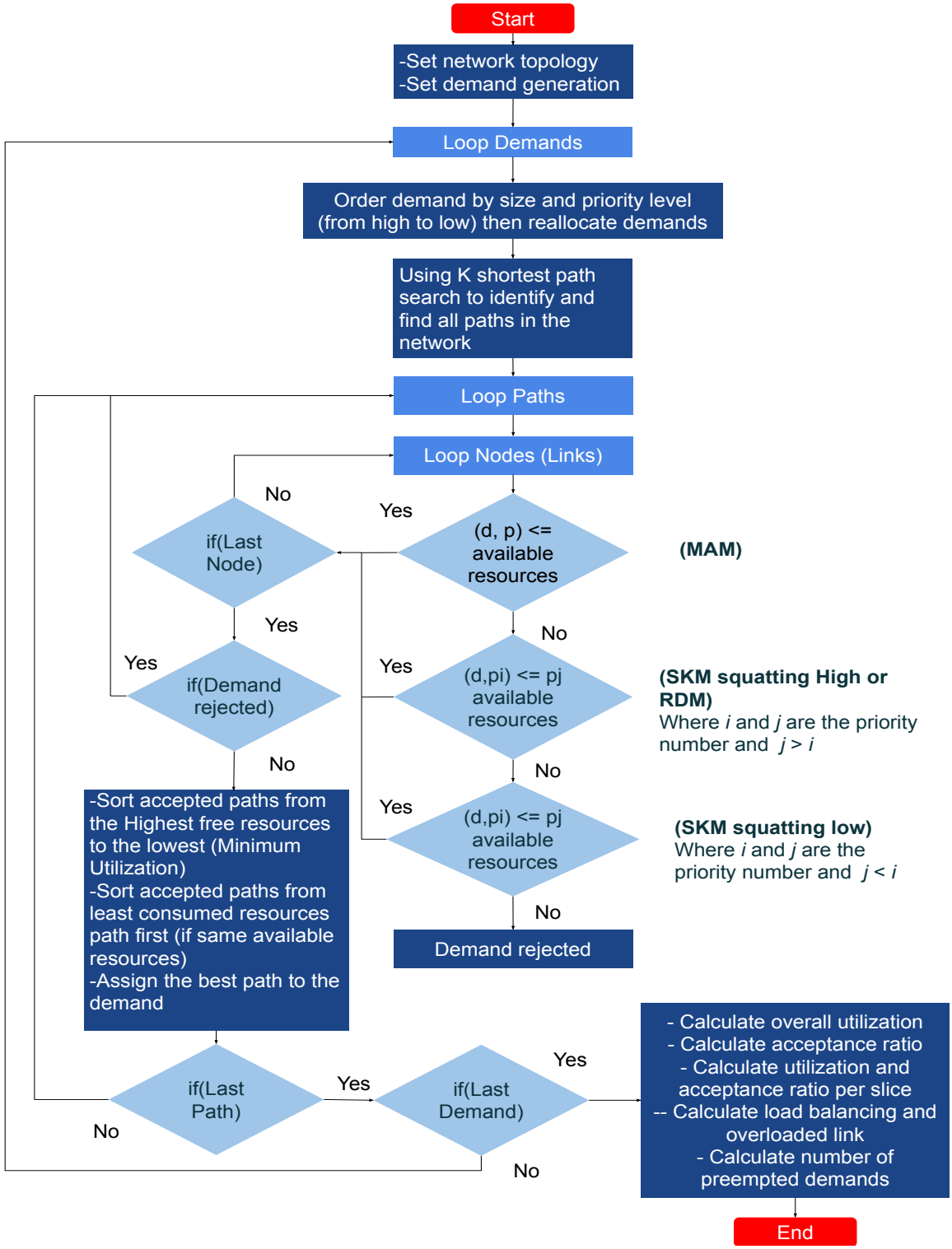


Figure 4.2: Flowchart 1 presenting the general structure of the methodology used in the offline mode of our proposed deployment policy. It starts by the initialization and routing phase, followed by the allocating step, and concludes by the evaluation phase

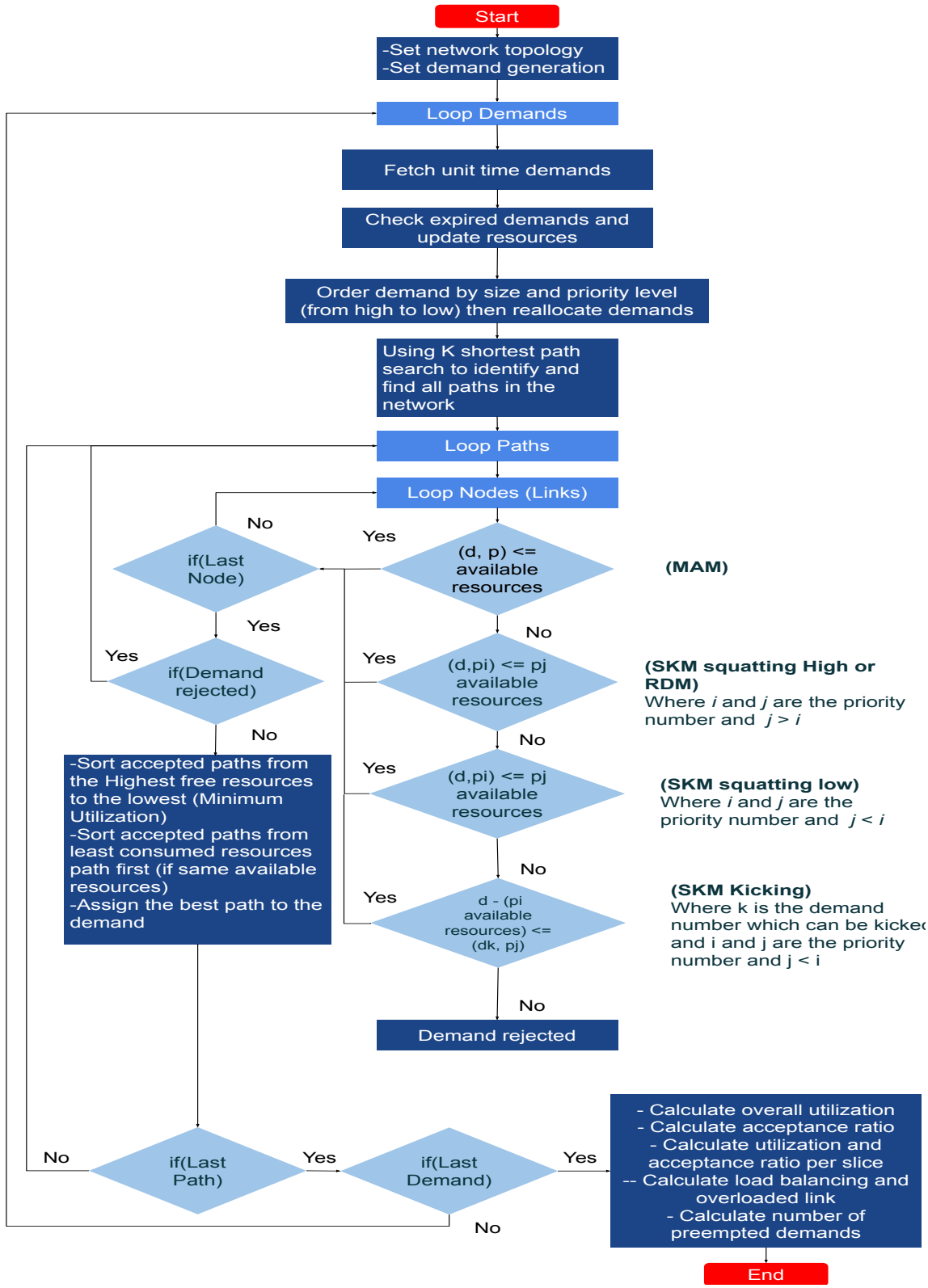


Figure 4.3: Flowchart 2 presenting the general structure of the methodology used in the online mode of our proposed deployment policy. It starts by the initialization and routing phase, followed by the allocating and updating phase, and concludes by the evaluation phase

4.4.3.1 Offline performance metrics

For the case of permanent demands (without lifetime), the total utilization (U), the utilization per slice (Uc), load balancing and overloaded link can be evaluated in Eq. (4.18-4.21) as below:
Please recall that,AR, AR_c, Bp, Bp_c equations are the same as those in Eq. (3.17-3.20)
Utilization, U**:**

$$U = \frac{1}{L} \sum_{\forall l \in L} \frac{R_z(l)}{R(l)} \tag{4.18}$$

Utilization per slice, U_c**:**

$$U_c = \frac{1}{L} \sum_{\forall CT_c(l) \in L} \frac{S_c(l)}{R(l)} \tag{4.19}$$

Load balancing, $LB(L)$**:**

$$LB(L) = \frac{\sum_{\forall l \in L} (U-\mu)^2}{|L|} \tag{4.20}$$

Overloaded link, L_{ov}**:**

$$L_{ov} = Max(U - \mu), \forall l \in L \tag{4.21}$$

4.4.3.2 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. (4.22) [Chowdhury (2012)]. Where the observation time of the system is from t until T.

$$AR(T) = \frac{1}{T} \sum_{\forall t \in T} \frac{AD(T)}{D(T)} \tag{4.22}$$

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.(4.23).

$$Bp(T) = \frac{1}{T} \sum_{\forall t \in T} \frac{BD(T)}{D(T)} \tag{4.23}$$

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.(4.24).

$$AR_c(T) = \frac{1}{T} \sum_{\forall t \in T} \frac{AD_c(T)}{D_c(T)} \tag{4.24}$$

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.(4.25).

$$Bp_c(T) = \frac{1}{T} \sum_{\forall t \in T} \frac{BD_c(T)}{D_c(T)} \tag{4.25}$$

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 [Kibalya (2020)] 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.(4.26).

$$U(T) = \frac{1}{T} \sum_{t \in T} \frac{1}{L} \sum_{\forall l \in L} \frac{R_z^t(l)}{R(l)} \tag{4.26}$$

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.(4.27).

$$U_c(T) = \frac{1}{T} \sum_{t \in T} \frac{1}{L} \sum_{\forall CT_c(l) \in L} \frac{S_c(l)}{R(l)} \tag{4.27}$$

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.(4.28) [Kibalya (2020)].

$$LB(L)(T) = \frac{\sum_{\forall l \in L} (U(T) - \mu)^2}{|L|} \tag{4.28}$$

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.(4.26) is the average resource utilization on the link l 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.(4.29).

$$L_{ov}(T) = Max(U(T) - \mu), \forall l \in L \tag{4.29}$$

4.4.3.3 Example of our proposed online deployment policy

In this example: The network topology consists of 6 Nodes and 9 links as illustrated in Fig. 4.4. 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

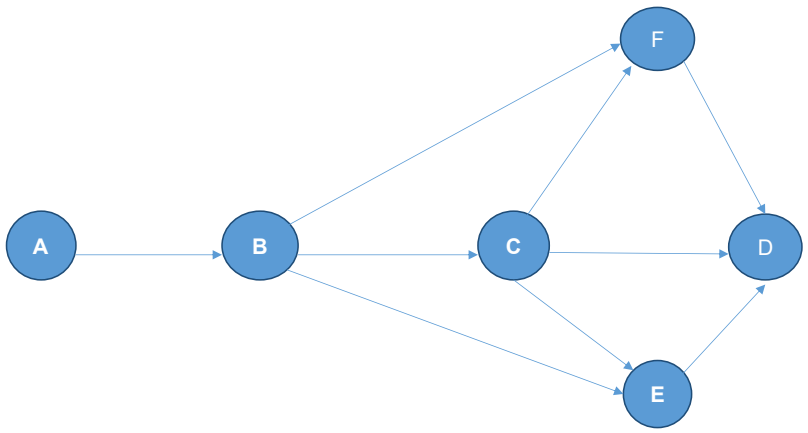


Figure 4.4: An illustration of the substrate network composed of six nodes and nine links in which the above four requests have to be mapped

Table 4.2 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 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 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 4.3, 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.

4.4.4 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 chapter:

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 [Reale (2016)], as explained in 4.4.5. 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

Table 4.2: 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**
$\#\mathbf{1:15_2(3)}$ **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**
$\#\mathbf{2:10_3(2)}$ **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**
$\#\mathbf{3:20_3(4)}$ **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**
$\#\mathbf{4:24_1(6)}$ **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

Table 4.3: 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 ($\#1 : 15_{2,3}(3)$)		

network topology and different generated traffic loads for traffic slices of all priorities are considered and described in 4.4.6. 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 4.4.7. 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 4.4.8. 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 4.4.9. The objective of this scenario is to analyze the robustness of SKM under permanent loading stress in the NSF topology.

4.4.5 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.

4.4.5.1 Simulation scenario settings

We adopted the settings of [Reale (2016)] 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 [Reale (2016)]). 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 4.4 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 4.5
- Exponentially modeled demand time life: average of 200 seconds (should cause link saturation)
- Simulation stop criteria: 1 h (3600 seconds).

Table 4.4: 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

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 4.4. Table 4.5 shows the phases of demand arrival. Phases 1 to 3 create a traffic profile where

Table 4.5: 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

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

4.4.5.2 Results evaluation

Fig. 4.5 shows the results obtained by each model in terms of U(T), AR(T) and P_{re}. As shown in Figs. 4.5a - 4.5c, 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. 4.5d 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. 4.5f - 4.5i). 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. 4.5e 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.

4.4.6 Scenario 2: Performance in online mode under mesh topology

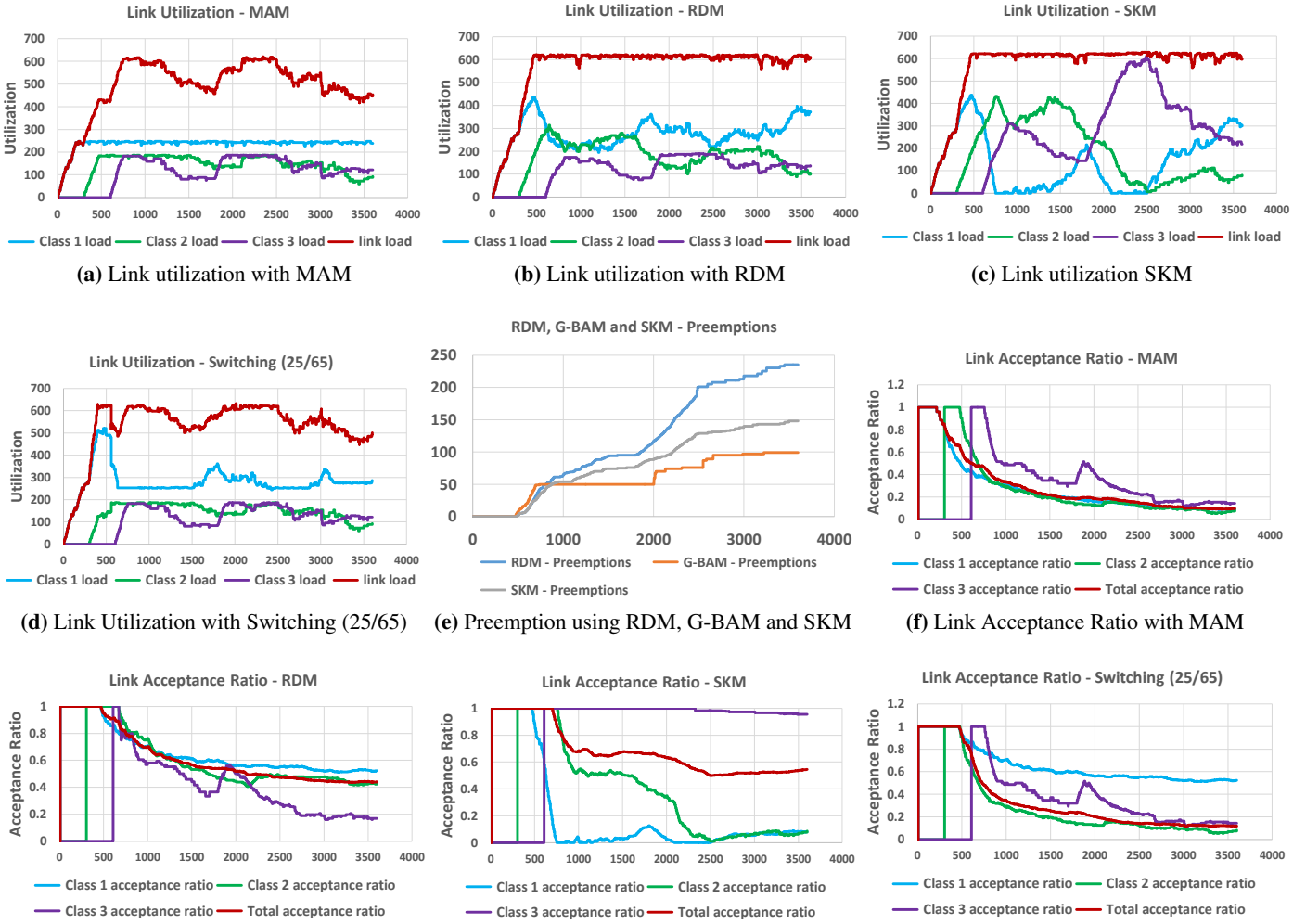
In this online scenario, we investigate SKM performance on limited resources of mesh network under different traffic loads and under fixed demands lifetime, in terms of several metrics, compared to MAM, RDM and AllocTC. Moreover, this scenario consists of three experiments. The main objective of the 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.

4.4.6.1 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. 4.6. 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.



(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 4.5: Comparison of utilization, preemption and acceptance ratio in first scenario

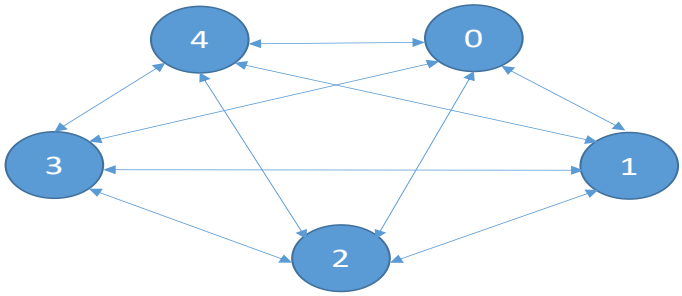


Figure 4.6: Mesh network topology

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.

Table 4.6: 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

4.4.6.2 Results evaluation

Table 4.6 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. 4.7- 4.8 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. 4.7g and Fig. 4.7h 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.

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 MAM by 18.02%, 18.34%, 22.54% in terms of U2 due to the kicking operation (see Fig. 4.7a). 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. 4.7d).

As shown in Fig. 4.8a, SKM, AllocTC and RDM resulted in 740, 739 and 745 respectively in terms of P_{re}. Moreover, Figs. 4.8d 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.

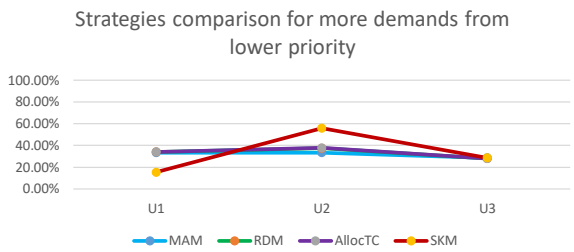
Considering same load in all priority slices: Fig. 4.7 illustrates that the SKM outperforms MAM, RDM and AllocTC in the highest priority slice by 20.47% in terms of U3 and 41.17% in terms of AR3. Also, the results show that SKM outperforms MAM, RDM and AllocTC in slice 2 by 11.94% in terms of U2 and by 17.39% in terms of AR2 (as the expected from the behaviors) due to the kicking operation as shown in Fig. 4.7b and Fig. 4.7e.

As shown in Fig. 4.8b, SKM outperforms AllocTC and RDM by 219 and 38 respectively in terms of P_{re} because of the kicking operation. Moreover, Figs. 4.8e 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.

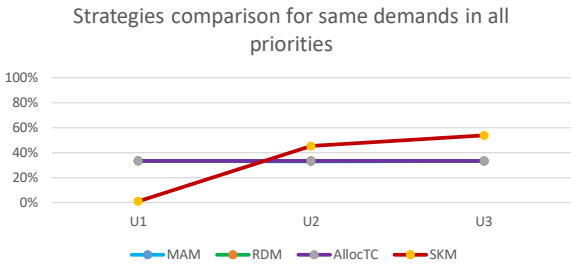
Considering high load in higher priority slices: Fig. 4.7c and Fig. 4.7f illustrate that the SKM outperforms AllocTC, RDM and MAM in the highest priority slice by 47.79%, 53.14% and 53.14% respectively in terms of U3 and by 54.28%, 60.95%, 60.95% respectively in terms of AR3. Moreover, the results of Fig. 4.8c 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. 4.8f 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.

4.4.7 Scenario 3: Performance in offline mode under mesh topology

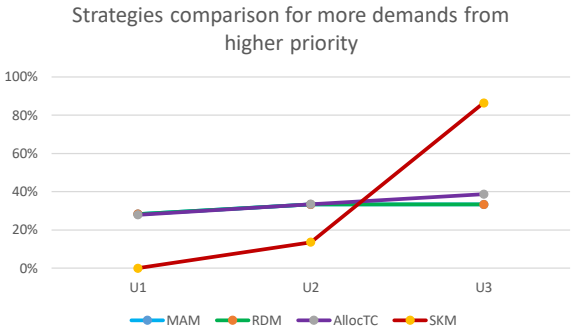
In this offline scenario, we used the same network topology and settings for the second scenario but with infinite demand lifetimes while considering a number of demands that vary from 201 to 2001 for each experiment in the studied scenario (see Table 4.7). Please note that the goal of this scenario is to investigate the robustness of SKM under constant load stress. Besides, SKM provides a good QoS level among different priority slices.



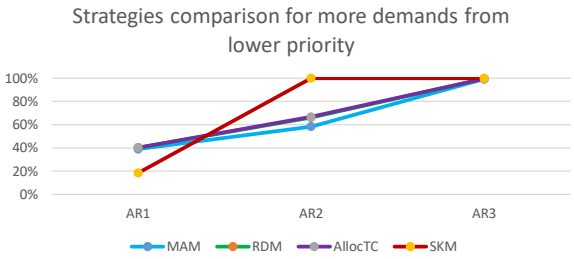
(a) Experiment 1: Average utilization per slice



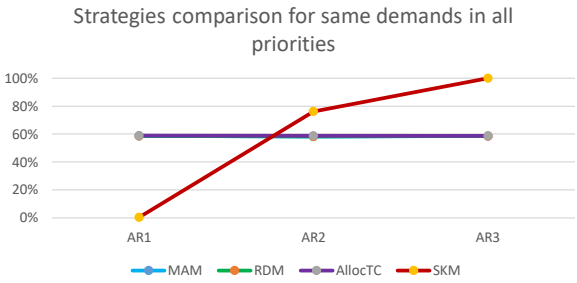
(b) Experiment 2: Average utilization per slice



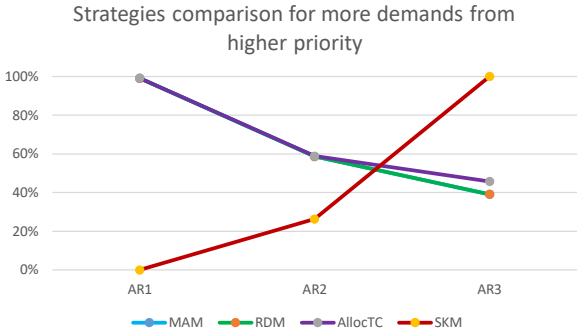
(c) Experiment 3: Average utilization per slice



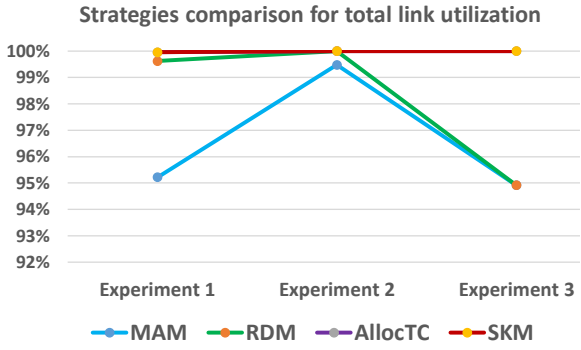
(d) Experiment 1: Average acceptance ratio per slice



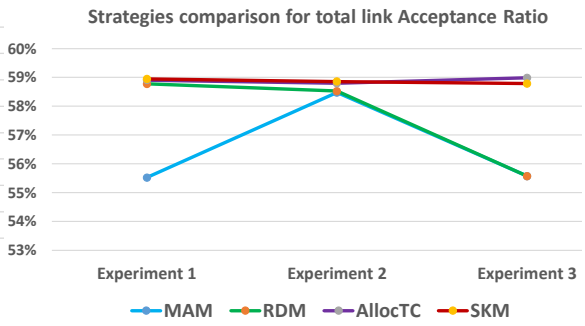
(e) Experiment 2: Average acceptance ratio per slice



(f) Experiment 3: Average acceptance ratio per slice

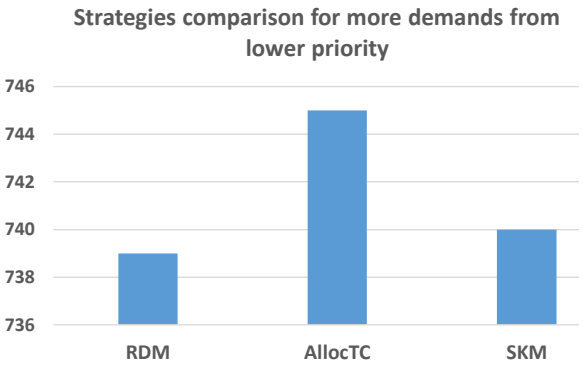


(g) MAM, RDM, AllocTC and SKM average utilization

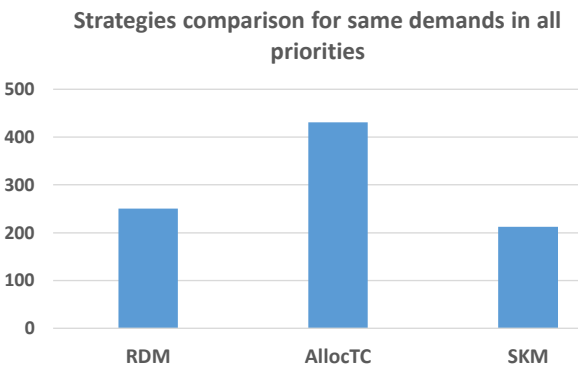


(h) MAM, RDM, AllocTC and SKM average acceptance ratio

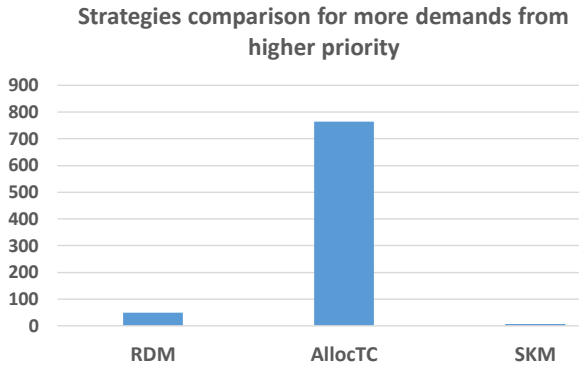
Figure 4.7: Comparison of utilization and acceptance ratio in second scenario



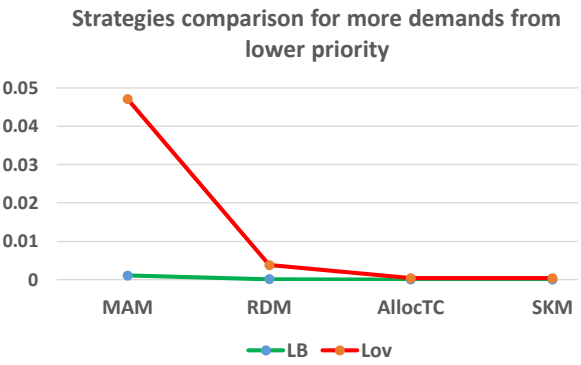
(a) Experiment 1: preempted demands by each strategy



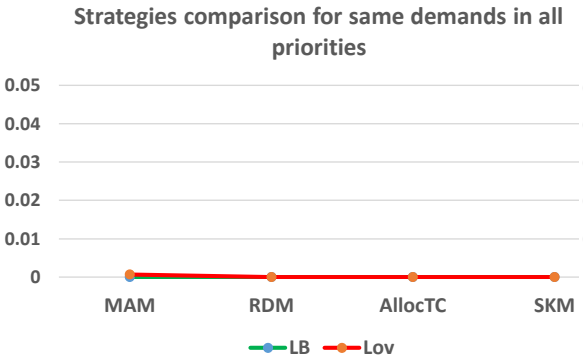
(b) Experiment 2: preempted demands by each strategy



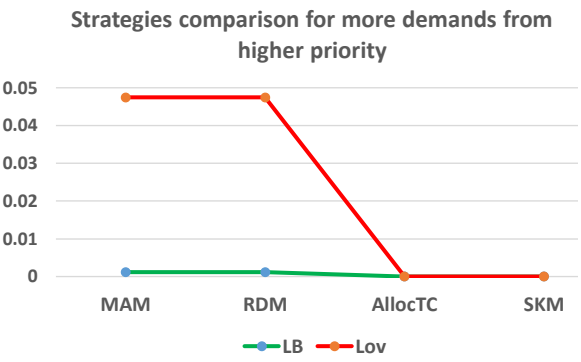
(c) Experiment 3: preempted demands by each strategy



(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 4.8: Comparison of preempted demands, load balancing and overloaded link in second scenario

4.4.7.1 Results evaluation

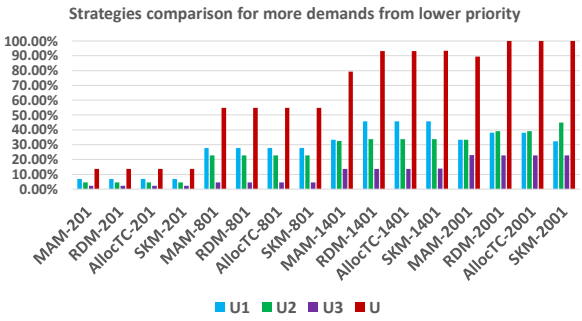
Figs. 4.9- 4.10 show the results of each algorithm in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov} using different traffic load according to experiments 1–3.

Table 4.7: Simulation experiments for the third scenario

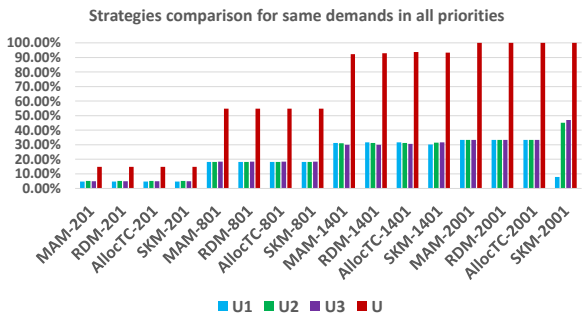
Number of slices in the generating file per-each unit time	Experiment 1 Load volume Traffic (Number of varied demands)	Experiment 2 Load volume Traffic (Number of varied demands)	Experiment 3 Load volume Traffic (Number of varied demands)
slice-Type 1	100, 400, 700, 1001	67, 267, 467, 667	34, 67, 201,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

Considering high load in lower priority slices: Fig. 4.9a and Fig. 4.9d 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. 4.10a- 4.10c 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. 4.10d 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.
Considering same load in all slices: Fig. 4.9b and Fig. 4.9e 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. 4.10d 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.
Considering high load in higher priority slices: Fig. 4.9c and Fig. 4.9f 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. 4.10f reveals that as demand size increases as

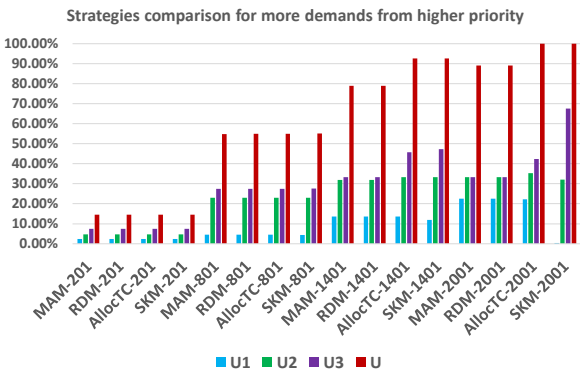
in the case where the size is 2001, SKM and AllocTC have similar performance in terms of LB and L_{ov} and had resulted in zero because all links are fully used across the network. On the other hand, MAM and RDM give the lowest performance in terms of LB and L_{ov} as they result in 0.0011 and 0.049, 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.



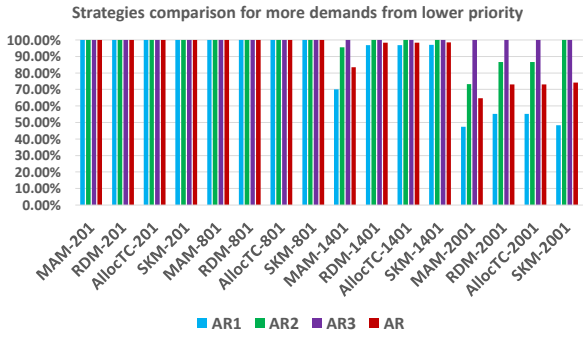
(a) Experiment 1: Total resource utilization



(b) Experiment 2: Total resource utilization

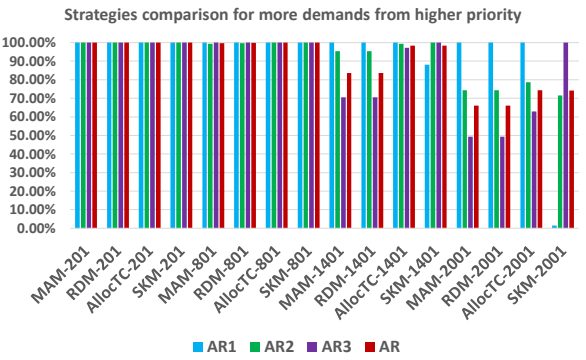


(c) Experiment 3: Total resource utilization



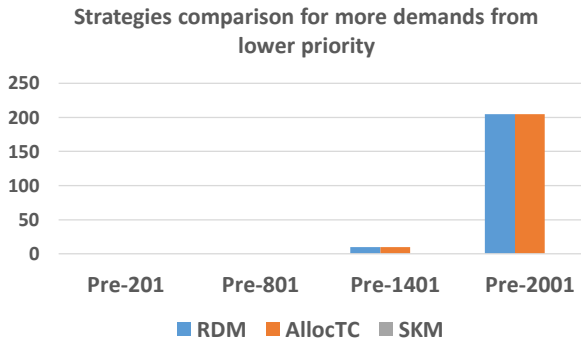
(d) Experiment 1: Acceptance ratio

(e) Experiment 2: Acceptance ratio

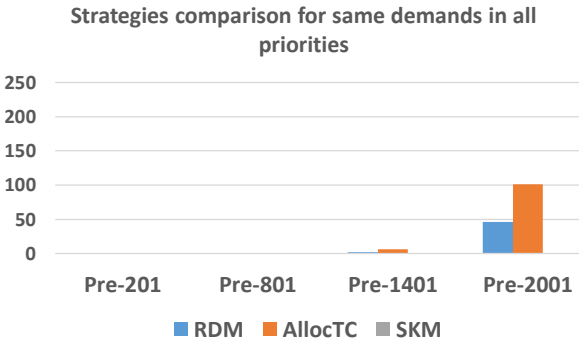


(f) Experiment 3: Acceptance ratio

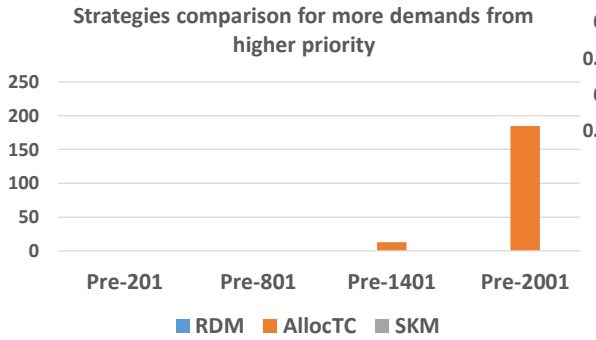
Figure 4.9: Comparison of utilization and acceptance ratio in third scenario



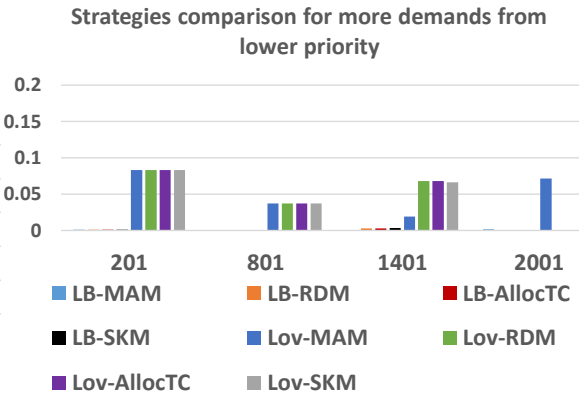
(a) Experiment 1: Preempted demands by each strategy



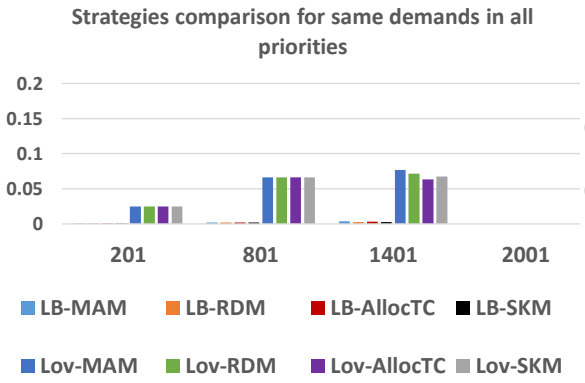
(b) Experiment 2: Preempted demands by each strategy



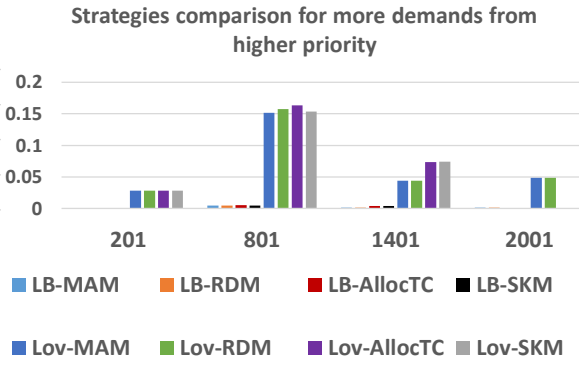
(c) Experiment 3: Preempted demands by each strategy



(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 4.10: Comparison of preempted demands, load balancing and overloaded link in third scenario

4.4.8 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.

4.4.8.1 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. 4.11. 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.

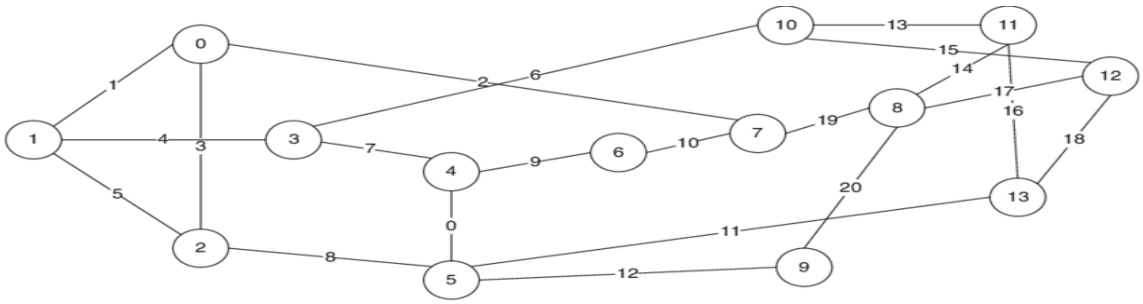


Figure 4.11: NSF topology

4.4.8.2 Results Evaluation

Table 4.8 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. 4.12 - 4.13 show the results of each algorithm in terms of U, AR, U_c, AR_c,

Table 4.8: 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

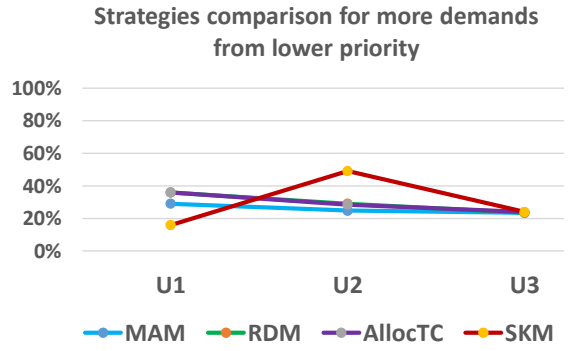
P_{re}, LB and L_{ov} using different traffic load according to experiments 1–3.
In terms of U and AR, Fig. 4.12g and Fig. 4.12h, illustrate the results from the three experiments for the MAM, RDM, AllocTC and SKM. **In the first experiment**, SKM, AllocTC and RDM result in

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.

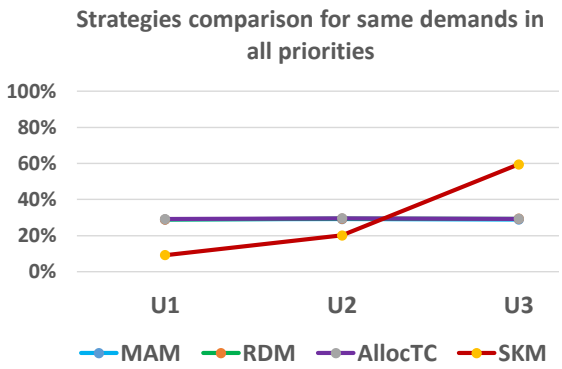
Considering high load in lower priority slices: SKM, AllocTC, RDM and MAM have similar behavior in terms of both U3 and AR3 as they both achieve 23.98% and 73.72%, respectively because of the load distributions on slice 3 across the network being lower than its capacity. Moreover, SKM outperforms AllocTC, 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. 4.12a and Fig. 4.12d). Furthermore, Fig. 4.13a 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. 4.13d 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.

Considering same load in all slices: Fig. 4.12 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. 4.12b and Fig. 4.12e. Moreover, Fig. 4.15b 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. 4.13e 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.

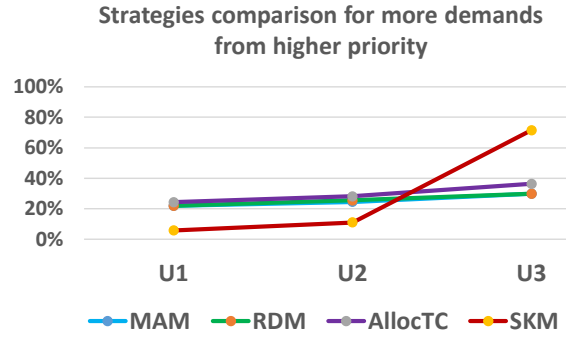
Considering high load in higher priority slices: Fig. 4.12c and Fig. 4.12f illustrate that the SKM outperforms AllocTC, RDM and MAM in the highest priority slice by 35.17%, 41.71% and 41.71% in terms of U3 and by 23.8%, 33.19% and 33.19% respectively in terms of AR3 (as the expected from the behaviors) due to the kicking operation. Moreover, from the results of Fig. 4.13c, 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. 4.13f 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.



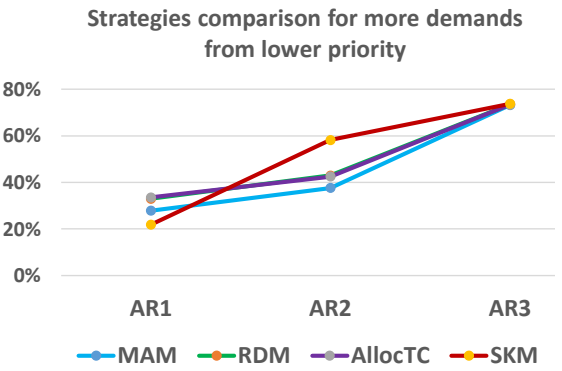
(a) Experiment 1: Average utilization per slice



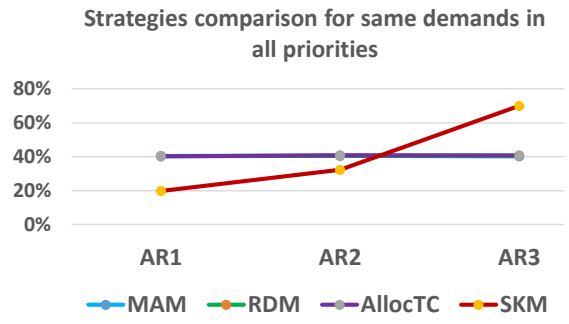
(b) Experiment 2: Average utilization per slice



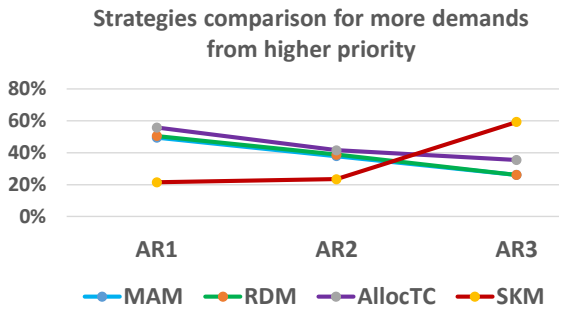
(c) Experiment 3: Average utilization per slice



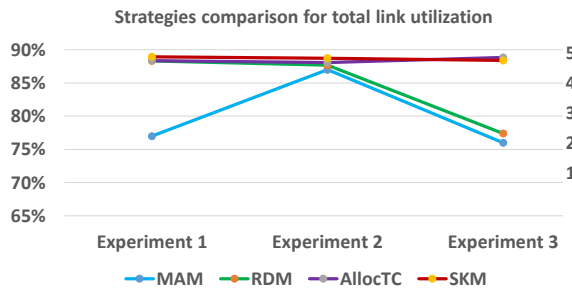
(d) Experiment 1: Average acceptance ratio per slice



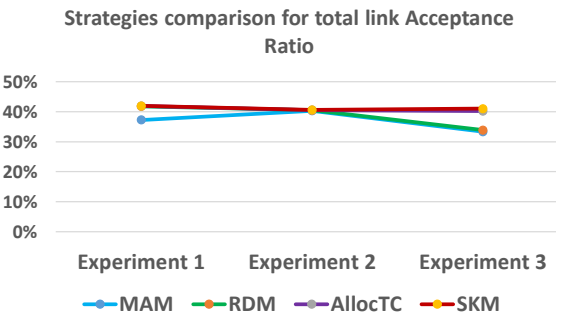
(e) Experiment 2: Average acceptance ratio per slice



(f) Experiment 3: Average acceptance ratio per slice

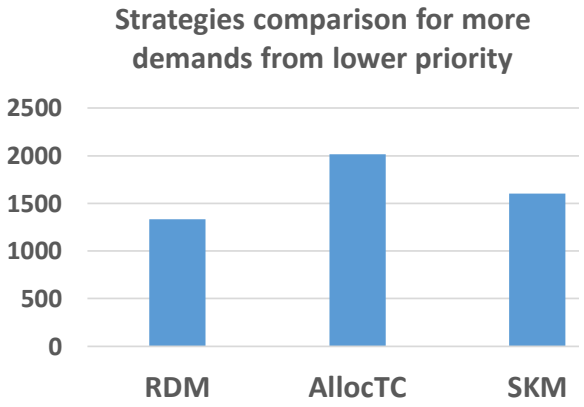


(g) Experiment 3: MAM, RDM, AllocTC, SKM utilization

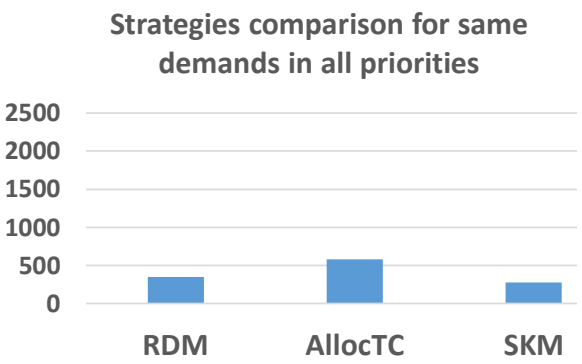


(h) Experiment 3: MAM, RDM, AllocTC, SKM acceptance ratio

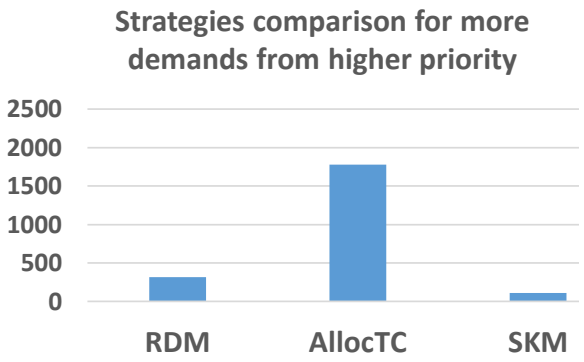
Figure 4.12: Comparison of utilization and acceptance ratio in fourth scenario



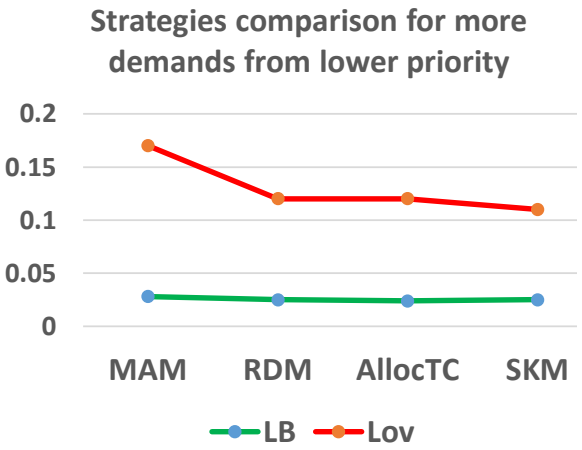
(a) Experiment 1: Preempted demands by each strategy



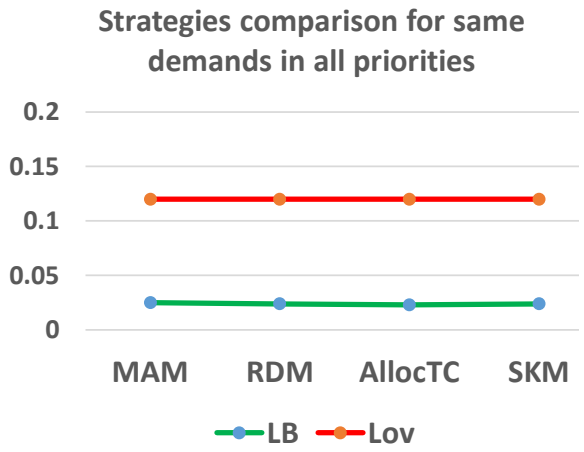
(b) Experiment 2: Preempted demands by each strategy



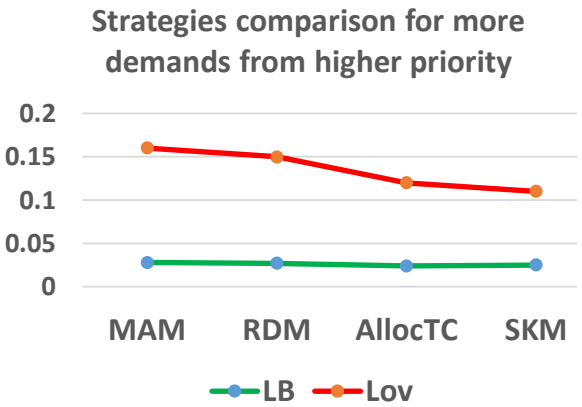
(c) Experiment 3: Preempted demands by each strategy



(d) 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 4.13: Comparison of preempted demands, load balancing and overloaded link in fourth scenario

4.4.9 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 4.9).

Table 4.9: 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

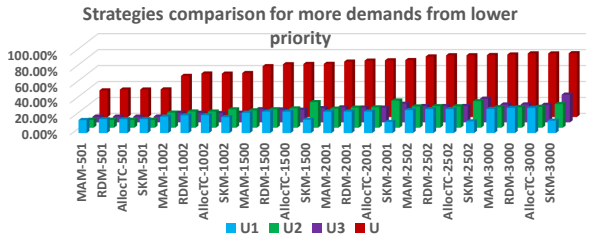
4.4.9.1 Results evaluation

Figs. 4.14- 4.15 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.

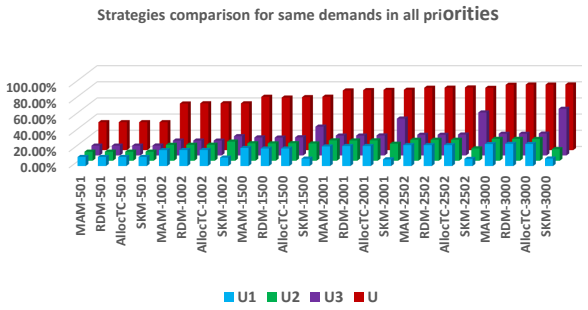
Considering high load in lower priority slices: Fig. 4.14a and Fig. 4.14d reveal that as demand size increases as in the case when the size is 3000, SKM outperforms the other algorithms in terms of both U3 and AR3 by achieving 12.38% and 27.2% respectively. Moreover, SKM outperforms AllocTC, RDM and 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. 4.15a- 4.15c 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. 4.15d 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.

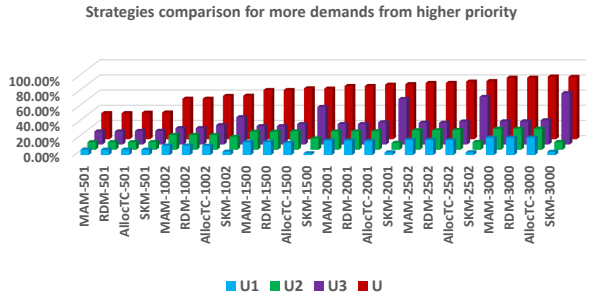
Considering the same load in all slices: Fig. 4.14b and Fig. 4.14e reveal that as demand size increases, as in the case where the size is 3000, SKM outperforms the other models in terms of both U3 and AR3 by 30.61% and 38.9%, respectively due to the sorting operation of SKM. Moreover, in terms of U and AR, all algorithms have similar 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. 4.15d 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.



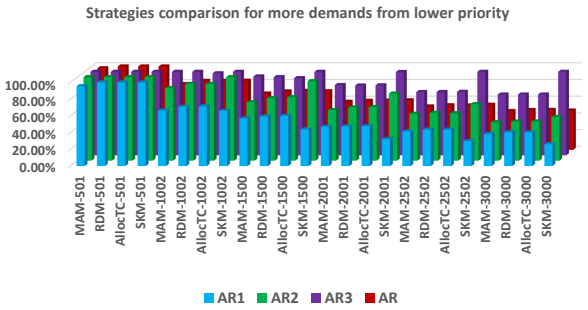
(a) Experiment 1: Total resource utilization



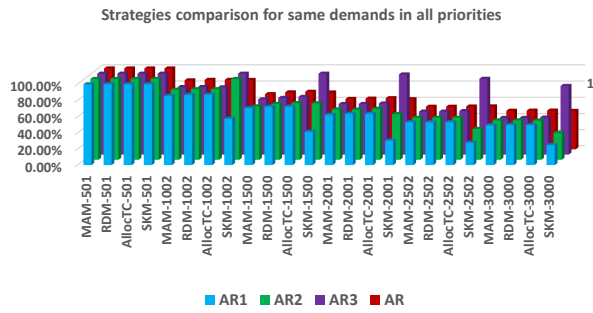
(b) Experiment 2: Total resource utilization



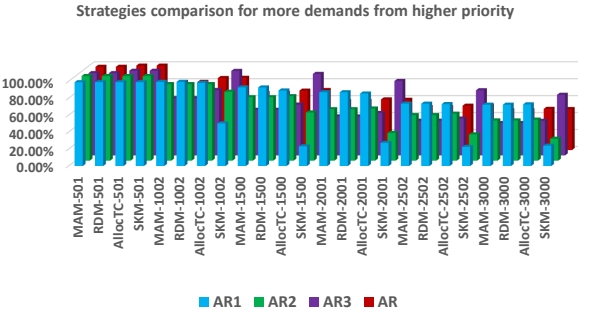
(c) Experiment 3: Total resource utilization



(d) Experiment 1: Acceptance ratio



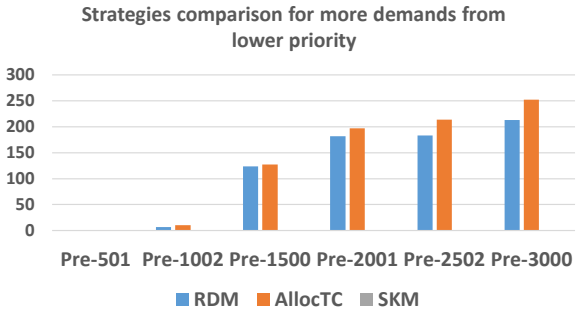
(e) Experiment 2: Acceptance ratio



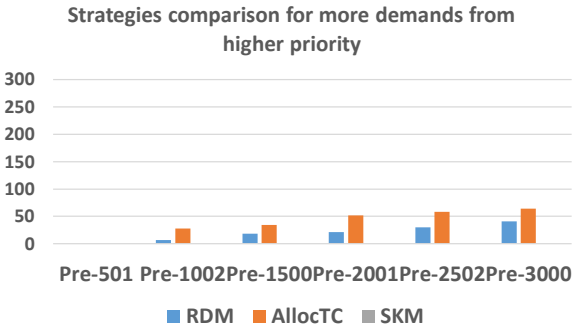
(f) Experiment 3: Acceptance ratio

Figure 4.14: Comparison of utilization and acceptance ratio in fifth scenario

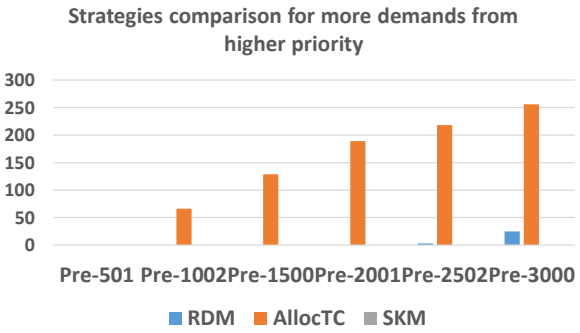
Considering high load in higher priority slices: Fig. 4.14c and Fig. 4.14f reveal that as demand size increases, as as in the case where the size is 3000, SKM outperforms AllocTC, RDM and MAM in terms of AR3 by 35.68%, 37.14% and 37.14% and by 30.8%, 33.4% and 33.4% respectively in terms of AR3 due to 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. 4.15f 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.



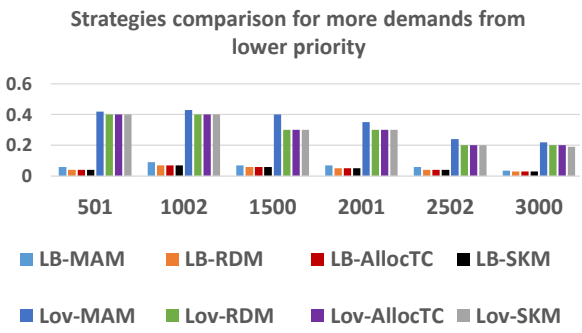
(a) Experiment 1: Preempted demands by each strategy



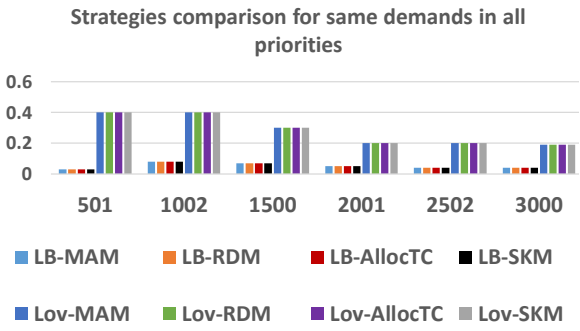
(b) Experiment 2: Preempted demands by each strategy



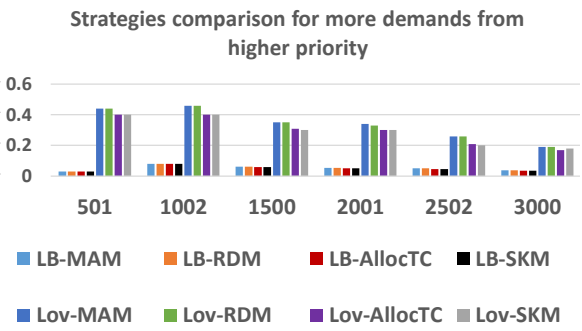
(c) Experiment 3: Preempted demands by each strategy



(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 4.15: Comparison of preempted demands, load balancing and overloaded link in fifth scenario

4.4.10 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:

4.4.10.1 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. 4.7b and Fig. 4.7e). 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.

4.4.10.2 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. 4.7b, Fig. 4.7e, Fig. 4.12b and Fig. 4.12e).

4.4.10.3 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. 4.12g, Fig. 4.12h, Fig. 4.12b and Fig. 4.12e). 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.

4.4.10.4 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.

4.4.10.5 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.

4.5 Conclusions

The chapter 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.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

CHAPTER 5

Evaluating the Impact of Delay Constraints on Network Services for Delay-Sensitive 5G Applications Based on SKM Model

5G and coming generations are envisaged to support a myriad of services with stringent requirements in terms of bandwidth and latency in different slices in order to provide a high-quality user experience. Therefore, intelligent resource allocation algorithms shall not only result in efficient utilization of network resources, but also guarantee the required quality of service for the priority slices. Moreover, cognizant of the strict latency requirements of the different services, such algorithms should include delay constraints of requests. As a contribution to this challenge, this chapter analyzes the impact of delay constraint on the performance of an online resource allocation algorithm based on an intelligent efficient squatting and kicking model (SKM), proved in Chapter-3 and Chapter-4 to be the most effective up to the present time yet. We formally define the proposed algorithm to solve the problem of real-time resource allocation for QoS end-to-end routing considering realistic 5G network behaviour by incorporating delay constraints and considering full network topology under online request arrival. In addition, this work extensively analyses the impact of delay on the performance of the proposed algorithm to introduce a new efficient deployment algorithm based on SKM, Moreover, the benefits of the new proposed algorithm will be reflected on creating real-time demands for 5G applications that are sensitive to delay, in addition to solving the resource allocation problem for large scale networks, using fewer resources and incurring lower costs.

5.1 Introduction

As previously discussed in Chapter 4, the 3rd Generation Partnership Project (3GPP) has identified enhanced Mobile Broadband (eMBB), massive Internet of things (MIoT), Ultra-Reliable and Low-Latency Communication (URLLC) and Vehicle-to-everything (V2X) as the four critical usage scenarios in 5G communication systems [3GPP1 (2020)], [3GPP2 (2020)]. Table 5.1 shows the significance, standard value and attributes used for each slice type.
Thus, in the 5G networks scenario, both high-bandwidth and low latency connectivity are a necessity. In order to cope with the ever-increasing traffic burden, these systems will need to be supported by intelligent resource allocation approaches for planning network deployments. Regardless of the used

Table 5.1: The significance, Standard value and parameters used for each slice type

slice/service type	Standard Slice Type (STT) Values	Characteristics	Slices attributes
eMBB	1	Slice suitable for the handling of 5G enhanced Mobile broadband, useful, but not limited to the general consumer space mobile broadband applications including streaming of High Quality Video, Fast large file transfers etc. It is expected this SST to aim at supporting High data rates and high traffic densities.	1- Connection density: [Kanavos (2021)] - The eMBB supports device density ranging from W_u = 200 to 2500/Km^2 - The URLLC supports device density ranging from W_y = 10 to 3000/Km^2 - The MIoT supports device density about W_e = up to 1million/Km^2 - The V2X supports device density about W_o = up to 1million/Km^2 $W_u \leq W_y$, $W_u <<< W_e$, $W_u <<< W_o$. 2- Data rate requirements: [Brown (2016)] - The eMMBB offers very high data rate: 10 to 20 Gbps - The URLLC provides low to medium data rates: about 50 kbps to 10 Mbps - The MIoT supports low data rates: about 1 to 100 Kbps - The V2X supports low data rates: up to 1Gbps $d_{w_u}(CT_c) >> d_{w_y}(CT_c)$, $d_{w_u}(CT_c) >>> d_{w_e}(CT_c)$, $d_{w_u}(CT_c) \geq d_{w_o}(CT_c)$. 3- Maximum acceptable E2E delay: - The eMBB is not restricted with delay constrains [Ijaz (2016)] - The URLLC offers 5 ms end to end latency between UE (i.e. mobile) and 5G eNB (i.e. base station) - The MIoT supports end to end latency from seconds to hours - The V2X supports about 3 to 100 ms end to end latency $\delta_{s,r}(l)_{W_u} >>> \delta_{s,r}(l)_{W_y}$, $\delta_{s,r}(l)_{W_u} >> \delta_{s,r}(l)_{W_e}$, $\delta_{s,r}(l)_{W_u} >> \delta_{s,r}(l)_{W_o}$.
URLLC	2	Supporting ultra-reliable low latency communications for applications including, industrial automation and (remote) control systems.	1- Connection density: $W_y <<< W_e$, $W_y <<< W_o$. 2- Bandwidth requirements: $d_{w_y}(CT_c) << d_{w_e}(CT_c)$, $d_{w_y}(CT_c) << d_{w_o}(CT_c)$. 3- Maximum acceptable E2E delay: $\delta_{s,r}(l)_{W_y} \geq \delta_{s,r}(l)_{W_e}$, $\delta_{s,r}(l)_{W_y} \geq \delta_{s,r}(l)_{W_o}$.
MIoT	3	Allowing the support of a large number and high density of IoT devices efficiently and cost effectively. MIoT services may impose any combination of bandwidth and delay according to devise types and deployment scenarios.	1- Connection density: $W_e \leq W_o$. 2- Bandwidth requirements: $d_{w_e}(CT_c) \leq d_{w_o}(CT_c)$. 3- Maximum acceptable E2E delay: $\delta_{s,r}(l)_{W_e} \leq \delta_{s,r}(l)_{W_o}$.
V2X	4	Slice suitable for the handling of V2X services, useful, but not limited to the Autonomous driving, Tele-operated driving and vehicular infotainment applications. It is expected this SST to aim at supporting massive numbers of ultra-reliable, low latency, high bandwidth communications.	-

algorithm, the allocation of the resources is constrained by link bandwidth as well as application delay requirements. Accordingly, not considering all necessary constraints during the resource allocation process will most likely result in degrading the overall quality of the whole allocation process [Xu (2019)]. Therefore, these approaches must be adaptable to various constraints while achieving possible maximum productivity and facilitation of sharing resources among slices while allowing a specific slice to meet the Service Level Agreement (SLA).

With the fast evolution of real-time transmissions in 5G networks, delay analysis is gaining attention in the literature [Tomovic (2016), Pateromichelakis (2017), Feng (2016)]. However, the analysis of delay in physical multi-hop paths is more challenging compared to physical single-hop paths in network slicing scenario, especially during disaster events and network congestion [El-mekkawi (2020)]. This is attributed to many factors impacting the end-to-end (E2E) delay in physical multi-hop paths networks, such as routing algorithm, network topology, traffic demand, and priorities. Consequently, the main motivation of this chapter is to introduce a new efficient deployment algorithm to solve the problem of real-time resource allocation, considering bandwidth, priorities and E2E delay in a multi-slice network based on Squatting and Kicking model (SKM), while aiming to maximize the overall resource utilization in the substrate network. The proposed online deployment algorithm allocates the available resource to different priority demands from source node to destination node along the routed path according to more realistic constraints, such as links' bandwidth and E2E delay. Moreover, the advantages of the proposed smart algorithm will be beneficial in creating real-time demands for delay-sensitive 5G applications, as well as solving the problem of allocating resources to large scale networks, using fewer resources and incurring fewer costs.

In line with bandwidth resource management in a multi-slice scenario, BAMs provide improved metrics compared to best-effort models. SKM outperforms the others by far especially, during congested scenarios, as shown in Chapter-3 and Chapter-4. Thus, this algorithm has been considered in this chapter.

In light of that, in this chapter, we analyzes the impact of delay constraint on the performance of an service deployment algorithm uses the intelligence of SKM strategy, defined in Chapter-3. SKM incorporates kicking and squatting of resources as innovative techniques enabling it to achieve 100% resource utilization and acceptance ratio for higher priority slices across the substrate network. The performance of the proposed algorithm is compared against other algorithms that use bandwidth allocation algorithms such as AllocTC. Finally, this chapter additionally provides analysis regarding the impacts of delay constraints on the performance of the proposed algorithm, further analysis on resource utilization and acceptance ratio in different network complexity scenarios.

Main contributions:

1. In-depth analysis of the impact of delay constraints on the performance of the proposed online algorithm, which represents the direct applicability of network slices on future 5G networks and beyond.
2. A formal definition of the proposed algorithm is introduced considering a real-time application for full network topology with delay constraints request.
3. A mathematical model for managing multiple network slices, which vary in terms of QoS

requirements.

4. Evaluations of acceptance ratio and utilization of substrate network links for the proposed online algorithm versus the most recent reference online algorithm were performed by Bahnasse et al. (2018) [Bahnasse (2018)].
5. Additional assessments of the proposed algorithm were also presented, illustrating the impacts of E2E delay against the state of the art algorithms while considering several metrics and scenarios under online cases.
6. Performance evaluations and analyses of the proposed algorithm are presented against routing algorithms incorporating BAM strategies in terms of several metrics reflecting the ability to manage multi-slice requests in a resource-limited 5G network reflecting the ability to accommodate various input traffic loads as well as the lifetime of requests.

A practical evaluation scenario:

In real 5G networks, the services, applications, and users need to interact with the infrastructure network directly and in real-time [Hejja (2018)]. Hence, in the context of network slicing, the requests must be analyzed and assigned on the physical substrate network, online, using the shared resources effectively, adhering to the necessary service qualities as required. Simultaneously, it is 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. To this end, the proposed deployment algorithm of this chapter is planned for an online scenario, and its main aim is to successfully allocate the requests among different priority slices, on real-time, while maximizing the total resource consumption in the entire substrate network considering E2E delay as the primary allocating constraint. The algorithm manages the network demands sequentially and continues observing and updating the substrate network, to allow more resources to be utilized in the future by new demands.

5.2 Formulation of the online problem

Since the resource allocation problem deals with making decisions about an efficient utilization under limited physical substrate network resources, the resource allocation problem was traditionally modelled as an optimization problem of an objective function, and constrained by controlling conditions, matching the availability of the resources against the requirements, while using the limited physical resources.

In this work, the problem is formulated with the link and slice constraints, subject to the priority service demands, network capacity and link resources (e.g., bandwidth). The inputs to the resource allocation phase are slice traffic, network capacity and substrate link resources. The output is the optimal deployment path for simultaneous slices demands that maximizes the network resources utilization while ensuring high admission for higher priority slices based on SKM. In this regard, the optimization of usage has two considerations. Network capacity is the primary consideration for describing the maximum number of resources that can be provided for forwarding traffic, which also plays a critical role in network load-balancing. Besides, the second consideration, we also take

into account the deployment propagation distance of forwarded traffic in terms of the substrate link resources.

Note that in this study, to facilitate the assessment of the proposed deployment algorithm, we applied only three types of slices discussed below, and also assume that service demand is acceptable when link resources are available along the required routing path from the source node to the destination node. The slices considered in this work are: 1) eMBB slice: This kind of slice is not strict with specific QoS requirements. Therefore, we assumed that this slice meets the lower priority service demands; 2) MIoT slice: This type of slice is characterized by a very large number of connected devices typically transmitting a relatively low volume of non-delay sensitive data. We assumed that this slice satisfies the intermediate priority service demands and 3) uRLLC slice: This type of slice is strict with the delay requirements. We assumed that this slice satisfies the highest priority service demands.

5.2.1 Infrastructure network model

We model the provided physical substrate network as a directed graph $G(X, L)$ where X and L indicate the set of all substrate nodes and substrate links respectively. Whenever such a connection exists, we indicate by $l \in L$ as a single substrate link between substrate node i $\in X$ and substrate node j $\in X$. Each substrate link l is described by i) Maximum link resources capacity R(l); ii) Available link resources at a given time denoted by $R_a^t(l)$; iii) Consumed link resources $R_z^t(l)$ at time t; iv) A set of traffic slices assigned along the link are denoted by $(CTs)(l)$, where $CT_N(l)$ is the highest priority slice and $CT_1(l)$ is the lowest priority slice; v) Actually allocated resources to slice c $S_c(l)$, where $c \in [1, N]$; vi) Slice resource constraints $RC_c(l)$; vii) Propagation delay $\delta(l)$. Whenever such a path exists, we denote by $P_{s,r}^n(l)$ as a possible physical path between substrate node (source node) s $\in X$ and substrate node (destination node) r $\in X$, the nth path between substrate node s and substrate node r for all $n \in [1, P_{set}^{s,r}(l)]$.

5.2.2 Slice demand model

Each demand belonging to any type of slice to be served in the network is defined 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 $c_{d_w} \in [1, N]$; VI) Maximum acceptable E2E delay $\delta_{s,r}(l)$ and v) lifetime interval t_{d_w}. Besides, we consider in this work that the size of the demand can be translated into a demanded number of link resources, so candidate paths from source to destination are calculated when the demand has reached.

5.2.3 Formulation of the online objective function

The principal purpose of SKM is to successfully accommodate all arriving demands on the online scenario, while maximizing overall resource utilization in the whole substrate network, by effectively allocating available resources for service demand in physical network paths. Demands in the online mode arrive with duration time t_{d_w}, therefore, the objective function must consider allocating demands

during the time intervals specified by each related demand. Thus, the main objective can be expressed as:

$$Max\ \mathbf{U}(\mathbf{T}) \tag{5.1}$$

Where U(T) is the utilization of the links across the network in each time window T. The link resource utilization relates to the ratio of the used link resources to the link capacity averaged over all substrate links and defined as follows:

$$U(T) = \frac{1}{T}\sum_{t\in T}\frac{1}{L}\sum_{\forall l\in L}\sum_{w\in W}\frac{X_w^{t,l} * d_w(CT_c)}{R(l)} \tag{5.2}$$

Where $X_w^{t,l} \in [1,0]$ is binary variable equal to 1 if the demand $w \in W$ is assigned resources at link $l \in L$, zero otherwise. $d_w(CT_c)$ denotes the demanded bandwidth resources by demand w. T denotes the duration of the simulation window in time units. The total consumed resources at edge $l \in L$ at any unit time t given by:

$$R_z^t(l) = \sum_{w\in W} X_w^{t,l} * d_w(CT_c) \tag{5.3}$$

This objective is subject to:

1. link constraints:

$$\sum_{\forall l\in P_{s,r}^n} RC_c(l) \leq R(l), \forall t, i, j \in X, i \neq j \tag{5.4}$$

$$\sum_{\forall i,j\in X} \delta(l) \leq \delta_{s,r}(l), \forall i, j \in X, \forall l_{s,r} \in X, i \neq j \tag{5.5}$$

$$\begin{aligned} Z(d_w) = R_a^t(l) \geq P_{s,r}^n(l) d_w(CT_c), \\ \forall t, \forall l \in P_{s,r}^n, w \in [1, D] \end{aligned} \tag{5.6}$$

   Eq. (5.4) guarantees that the maximum reservable bandwidth for a link l at any time is less than or equal to the link capacity of that link. E2E delay in $P_{s,r}^n(l)$ is controlled through Eq. (5.5) to be less than or equal to the maximum demanded delay by demand w at a given time. Eq. (5.6) characterizes if demand w was successfully assigned at a specified time on all links along the path that have more available resources than required.

2. Slice constraints:
   During lifetime interval t_{d_w}, to assign the demand into a set of traffic slices across the link, SKM executes four steps, which are formally defined in [El-mekkawi (2020)]. In brief these steps are: **Step 1:** Upon the arrival of a demand $d_w(CT_c)$ belonging to slice c, SKM begins acting as a normal MAM algorithm. If resources are not enough, try step 2. **Step 2:** SKM

checks where the resources are not used, starting with higher priority slices (Squatting-High ($SH_q(l)$) or RDM). If the resources are still insufficient or unavailable, in **Step 3** the algorithm attempts to share resources from lower priorities (Squatting-High-$SL_q(l)$). Finally, in **step 4** the algorithm becomes more aggressive, expelling lower priorities when no other options are possible (kicking operation-$K_q(l)$).

5.3 Deployment policy of network slicing based on SKM

In this section, we present the proposed deployment policy for allocation of priority demands in a multi-slice network. Mainly, we provide a comprehensive discussion of the different steps included in the execution of the algorithm.

The optimal solution of resource allocation, admission control and QoS management for multiple slice network requires smart algorithms in order to dynamically support, discover, and reserve limited network resources that are often different in type, implementation and priorities. One of the main reasons for the complexity of the network resource allocation problem is the random arrival of user requests and the limited substrate network resources. Nonetheless, the resource allocation problem is an NP-hard due to link allocation problem, since it is difficult to ensure that the routing paths meet the QoS constraints under limited network resources. Besides, the difficulty in selecting the optimal path for various priority requirements and subsequent allocation of resources from source to destination in the physical substrate network. Consequently, most resource allocation algorithms need to resolve resource allocation optimization problems on time. Thus, to resolve the objective function, this chapter introduces the online deployment algorithm, as a priority aware resource allocation algorithm, which can optimize the use of resources by effectively allocating different priority service requirements in terms of link resources based on SKM strategy across the entire network considering the following constrains, namely: link, slice and E2E delay. If the physical path links contain sufficient resources to provide the resources required for the request, a successful allocation occurs.

5.3.1 Description of the proposed deployment policy

The methodology of the proposed deployment policy is described in the flowchart shown in Fig. 5.1. Also, The algorithm pseudo-code is specified in the Algorithm 5. At each time t, when a request arrives, the deployment process will perform four significant steps to allocate the demand:

1. **Step 1 (Routing algorithm)**: Find all possible paths from the source to destination to allocate the demand in the network.
2. **Step 2 (Resource update)**: In each time unit, before executing the resource allocation process, the algorithm continues to determine whether any request is expired or not, to eliminate its requests from the hosting resources, and updates the entire substrate network accordingly.
3. **Step 3 (Allocating decisions)**: Check all potential paths according to the available resources metric defined using a specific allocation strategy.

- On each path (check the node for each node), check the required delay and resources for the

request. If the path delay and the resources available along this path meet the requirements of the request, add the path to the list of potential paths. Otherwise, ignore the path.

- Define a specific allocation strategy (SKM) to optimize individual node allocation based on efficient use. SKM allocates resources and control process to check whether network resources are sufficient to serve user demand resources and QoS requirements.

4. **Step 4 (Path selection strategy)**: Sort the potential paths according to the maximum resources available on the links and choose the best path to allocate the demand. If the available resources are the same in two or more paths, the potential paths will be sorted according to the path that consumes the least resources first.

More details on the steps of the proposed algorithm are discussed in the subsections.

5.3.2 Routing algorithm

The proposed algorithm starts by checking for all possible paths to determine the transmission path for the request from source to destination. In order to find all possible paths to assign service demand, many methods and strategies such as Brute force and others can be used for this task. Due to the complexity of the path calculation, we used the k Shortest Path (KSP) algorithm, which is used to determine the K shortest path where K is an integer number of shortest physical paths appropriate to satisfy the bandwidth resource requirements for various priority slices. Please note that our work assumes that all possible paths are checked first, and then optimal paths are chosen through checking node per node if there are sufficient resources by using the proposed (SKM) strategy according to the demands and network infrastructure. Moreover, the substrate network topology considered in the study is assumed to be constant. Thus, the main factors that express the substrate networks, for example, the quantity and connectivity of nodes and links in the substrate network, are also constant and do not change, but only the capacities of their resources differ due to their use after each time period t.

5.3.3 Resource update

In each unit time, the algorithm verifies the expiration of requests and substrate network resources before allocating resources to the subsequent request. In other words, the algorithm fetches a set of multiple requests sequentially from the request generation file (D) list and checks for the expiration of the allocated requests. Later checking the expiration stage, requests will be classified according to size and priority level from highest to lowest. Once the arrangement stage occurs, the process assignment of step 2 will be used to allocate requests along the network topology paths.

5.3.4 Embedding decisions

This is the essential step that ensures that every request in the network is allocated, which solves the problem of allocating resources along the required physical path. To ensure allocation, we first verify that the path delay meets the demand delay. Then, we use SKM strategy in the nodes along the path to optimizing requests in terms of bandwidth by exploiting the partition and reservation of resources

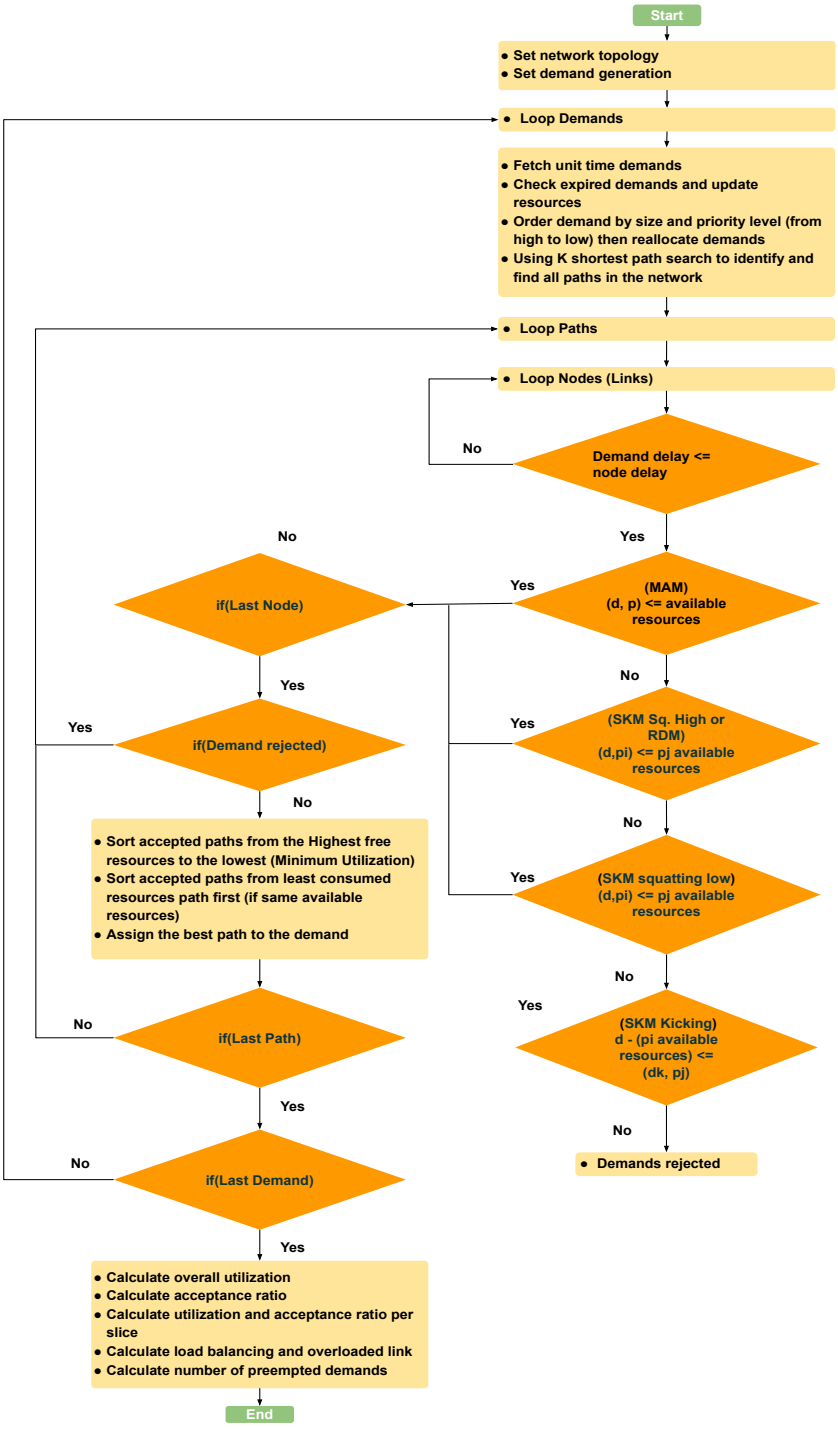


Figure 5.1: Flowchart 1 illustrates the methodology structure used by the proposed deployment policy. It begins with the routing step, then it is followed by the allocating and resource update steps, and it ends with the assessment step

Algorithm 5 Pseudo-code of the proposed deployment algorithm

Input: G[X, L] of X routers and L links, and set of service demands D to be allocated
Output: Allocation status, Z(d_w): Succeed, R: Reject
while $t \neq 0$ **do**
  $D_{selected} \leftarrow D_{(i-1)n+1:in}$ Fetch n demands consecutively from D $D_{checked} \leftarrow \Phi(D_{selected})$ Check Expiry of the Demands $D_{sorted} \leftarrow$ SortDemands($D_{checked}$) Sort Demands
  for *Each Demand* d_w = $d_w(CT_c) \in D_{sorted}$ *arriving the substrate network randomly at time* t **do**
    Initialize A as empty set
    Start SKM **assignment process** Loop D :Demands
    for *Each* $l \in P^n_{s,r}(l) \in K$ *shortest path list (Step 1)* **do**
      Ensure that the link delay meet Demand d_w delay using Eq. (5.5)
      Calculate available resources of a link l using Eq. (5.7) **if** *Demand* d_w *assignment process was successful for* $P^n_{s,r}(l)$ **then**
        Add $P^n_{s,r}(l)$ into A as potential path
      end
    end
    if *Count A* > 0 **then**
      for *Each* $P^n_{s,r}(l) \in A$ **do**
        Determine path available resources
        $R^t_a(l) \Leftarrow$ Min($R^t_a(l)$)
        for ($P^n_{s,r}(l)$ *and* $R^t_a(l) > 0$) **do**
          Select the optimal path based on highest available resources as Eq. (5.9)
          if *two paths or more have same amount of available resources along the path* **then**
            compute the least consumed resources path as Eq. (5.10)
            $Z(d_w)$: Succeed d_w for $P^n_{s,r}(l)$
            Evaluate Metrics using Eq. (4.22-4.29)
          end
        end
      end
    end
    else
      R: Reject d_w for $P^n_{s,r}(l)$
    end
  end
end

according to different priority slices and the flexibility to use the full amount of resources when no slice needs them. For instance, Fig. 5.2 describes the process of allocating a service request in the physical routing path based on SKM strategy. The service request w contains the substrate source node s and the substrate destination node r. The path (s, r) is the chosen physical routing path that is allocated by the service request W and contains two links s i and i r for transmitting traffic from the node s to node r. Specifically, in the time period t_{d_w}, the service request w is assigned to a priority slice defined by the SKM steps in each link along the path that satisfies the bandwidth and delay required by the service request. Requests are arranged according to size and priority to reduce the number of kicking actions per unit time and also to optimize resource usage in the network because arranging requests according to size resulting in higher utilization rate in most cases. In each time unit, SKM executing a sorting process before starting a new request allocation. However, if there are no resources available for all the candidate paths to accepting the request, the request will be rejected and move to the next request from step 1. This process ends when there are no other paths to accept the requests.

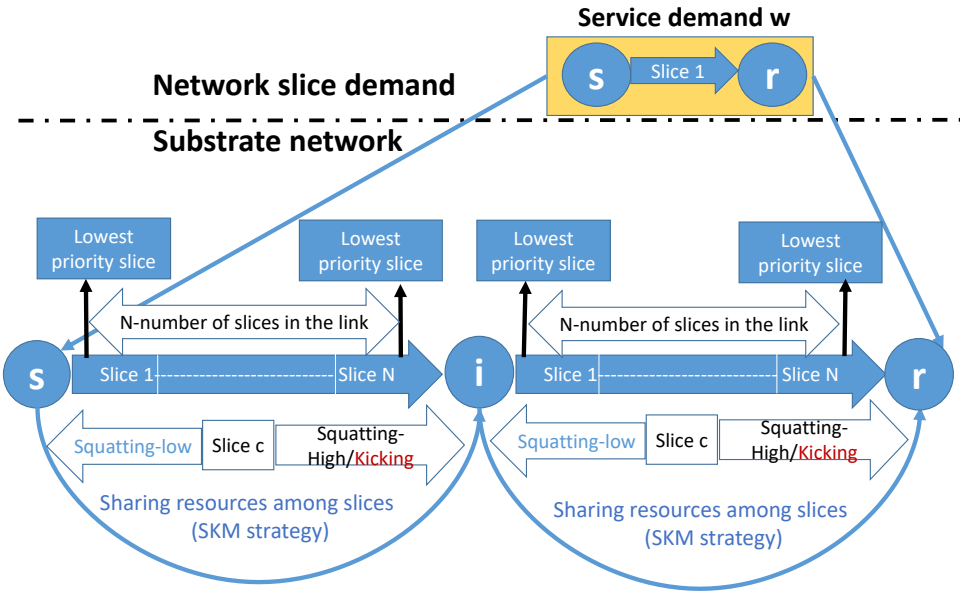


Figure 5.2: Illustration of link mapping along the path using SKM strategy

5.3.5 Path selection procedures

In this step, the algorithm will perform three procedures to select the optimal path from the list that contains all the potential routing paths that can accept the demand, which is provided using the defined resource allocation strategy (SKM).
More details on the three procedures are described below.
Procedure 1, at any time variable t, the algorithm will determine the available link resources as follows:

$$R_a^t(l) = R(l) - R_z^t(l) \tag{5.7}$$

$$R_z^t(l) = \sum_{c=1}^{N} R(l) - (RC_c(l) - min(S_c(l), RC_c(l))) \tag{5.8}$$

Eq. (5.7) illustrates the computation of available resources in a link. Furthermore, $R_z^t(l)$ is calculated by the summation of the difference among $R(l)$, and the minimum between allocated and reserved resources for each class as Eq. (5.8). Where N is the number of slices across the link l.
Next, the path available resources are the min value of $R_a^t(l) = Min(R_a^t(l))$.
Procedure 2, is to choose the best routing path considering the constraints of the resources of a link. The optimal routing path can be addressed concerning QoS constraints when its links meet resource constraints.
The links along the requested path meeting the constraints of the resources are described by Eq. (5.9):

$$Max\left\{Min(R_a^t(l)) \geq P_{s,r}^n(l) d_w(CT_c)\right\}, \quad \forall l \in P_{s,r}^n(l), n \in [1, P_{set}^{s,r}(l)] \tag{5.9}$$

Procedure 3, if more than one routing path has the same amount of available resources, the best path will be the one with the lowest amount of bandwidth consumed.
To compute the less consumed path Eq. (5.10).

$$Min\left\{\sum_{\forall l \in P_{s,r}^n(l)} R_z^t(l)\right\}, n \in [1, P_{set}^{s,r}(l)] \tag{5.10}$$

5.3.6 Evaluation metrics

The performance of the proposed deployment algorithm will be evaluated based on acceptance ratio, total resource utilization, load balancing, overloaded link and total number of preempted demands across the network. Please recall that, AR, AR_c, U, U_c, $LB(L)$, L_{ov} equations are the same as those in Eq. (4.22-4.29) [Chowdhury (2012), Fischer (2013), Kibalya (2020)].

5.3.7 General illustrative example

In this subsection, we give a brief numerical example to explain the proposed deployment algorithm considering a network topology (directed graph) consisting of 6 nodes and 9 links as shown in Fig. 5.3. Moreover, we assume that all links have the same capacity of 30 units and the same delay of 1ms. Every link is distributed into 3 priority slices and all slices have the same quantity of resources equal to 10 units. Furthermore, four demands (from one source to different destinations) need to be allocated based on the resources available across the network as described below. Please note that we considered the K shortest path algorithm with K equal to 2 for this example to map the different demands with the generation rate set to one demand per each unit time as follows:

#1: From A to D, 15 units, priority 2, delay 3ms, duration=3
#2: From A to E, 10 units, priority 3, delay 2ms, duration=2
#3: From A to F, 20 units, priority 3, delay 5ms, duration=4
#4: From A to F, 24 units, priority 1, delay 4ms, duration=6

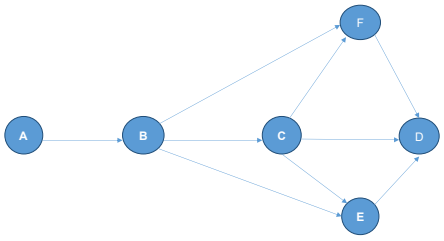


Figure 5.3: An illustrative diagram showing a physical network of 6 nodes and 9 links receiving three service demands

Table 5.2 explains the basis of our proposed deployment policy behaviour in the example shown above with an online mode in terms of resource allocation and reservation for the demand by considering traffic slices and link capacities. Moreover, the above example demonstrates SKM's ability to efficiently make decisions to determine paths according to priority and delay demands. The first column of the table shows the optimal computing paths for mapping four E2E service demands across substrate network per unit time. The second left column demonstrates the routing step of the deployment policy. Before starting the process of allocation in each unit time, the algorithm checks the expiry of allocated demands and the substrate network is updated as shown in column three (expired demands) and five (available resources in each link of path). For instance, before the arrival of demand $\#4 : 24_{1,4}(6)$, the demand $\#2 : 10_{3,2}(0)$ was expired and the available network resources was updated. Please note that the allocation of the demands is performed after the sorting process in each unit as shown in column six (Alive demands after sorting). For example, when the demand $\#4 : 24_{1}(6)$ arrives at the network, firstly, we must do rearranging including the new demand to the existing alive demands according to size and priority. Next, the demands $\#3 : 20_{2,5}(3)$ and $\#4 : 24_{1,4}(6)$ are allocated respectively. Also, in this example, we show how SKM uses kicking operation to favour the higher priority slices as in unit time 3. In unit time 3, there is no enough resources in the network to accept the new higher priority demand $\#3 : 20_{3,5}(3)$ so, the algorithm checks all alive demands that can be kicked to favour the new demand. Accordingly, demand $\# : 15_{2,3}(3)$ is expelled from the network to allocate new higher priority demand as shown in column seven (execution). Finally, after verifying all potential paths that can allocate the demand, the algorithm determines the optimal path based on the available resources, as shown in the last column from Table 5.2. Table 5.3 shows the results of the online proposed algorithm in terms of the U_c, U, AR_c, Bp, AR, LB(L) and L_{ov}. From the results, slice 3, accepted two demands until the observation time $\#2 : 10_{3,2}(2)$, $\#3 : 10_{3,5}(2)$ across the network. Please note that the low priority slice demand $\#1 : 15_{2,3}(3)$ has been kicked to satisfy the higher priority slice demand $\#3 : 20_{3,5}(2)$.

5.4 Simulation and analysis

In this section, the performance analysis of the proposed algorithm is discussed including the resource allocation algorithms, the different scenarios that are considered in this work, simulation settings and the obtained results.

5.4.1 Simulation scenarios and compared algorithms

Table 5.4 provides a high-level comparison between SKM, Smart Alloc, AllocTC, RDM and MAM algorithms, listing their used strategies, and how they embed the demands onto different priority slices along the path. 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 as follows:

1. Scenario 1: SKM's overall performance in terms of acceptance rate, resource usage, as well

Table 5.2: Numerical example showing the basics of our proposed deployment algorithm

#of demand: $d_{p,\delta}(t)$ & 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**
	$P_{A,B,C,D}$ in this path: [4]nodes and [3] links			A-B (10,10,10) B-C (10,10,10) C-D (10,10,10)		Accepted delay 3ms (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
$\#\mathbf{1}:\mathbf{15_2(3)}$ **The selected path after checked all paths is** $P_{A,B,C,D}$	$P_{A,B,F,D}$ in this path: [4]nodes and [3] links	-	$\#1:15_{2,3}(3)$	A-B (10,10,10) B-F (10,10,10) F-D (10,10,10)	-	Accepted delay 3ms (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**
	$P_{A,B,E}$ in this path: [3] nodes and [2] links			A-B (10,0,5) B-E (10,10,10)		Accepted delay 2ms (5,0,0) SL1_(A-B) (10,10,0) SL1_(B-E)	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
$\#\mathbf{2}:\mathbf{10_{3,2}(2)}$ **The selected path after checked all paths is** $P_{A,B,E}$ (the least consumed resources path)	$P_{A,B,C,E}$ in this path: [4] nodes and [3] links	-	$\#2:10_{3,2}(2)$	A-B (10,0,5) B-C (10,0,5) C-E (10,10,10)	$\#2:10_{3,2}(2)$ $\#1:15_{2,3}(2)$	Rejected delay 2ms	Discarded path due to rejected delay
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**
	$P_{A,B,F}$ in this path: [3] nodes and [2] links			A-B (5,0,0) B-F (10,10,10)		(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
$\#\mathbf{3}:\mathbf{20_{3,5}(4)}$ **The selected path after checked all paths is** $P_{A,B,F}$ **(the least consumed resources path)**	$P_{A,B,C,F}$ in this path: [4] nodes and [3] links	-	$\#3:20_{3,5}(4)$	A-B (5,0,0) B-C (10,0,5) C-F (10,10,10)	$\#3:20_{3,5}(4)$ $\#2:10_{3,2}(1)$ $\#1:15_{2,3}(1)$	(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**
	$P_{A,B,F}$ in this path: [3] nodes and [2] links			A-B (5,0,5) B-F (10,0,0)		Accepted delay 3ms (5,0,5) Rejected (10,0,0) Rejected	Discarded path due to rejected allocation
$\#\mathbf{4}:\mathbf{24_{1,4}(6)}$ **The demand is rejected**	$P_{A,B,C,F}$ in this path: [4]nodes and [3] links	$\#2:10_{3,2}$ (0)	$\#4:24_{1,4}(6)$	A-B (5,0,5) B-C (10,10,10) C-F (10,10,10)	$\#3:20_{2,5}(3)$ $\#4:24_{1,4}(6)$	Accepted delay 3ms (5,0,5) Rejected (0,0,6) RDM (0,0,6) RDM	Discarded path due to rejected allocation

Table 5.3: Results of the performance metrics after applying the proposed deployment algorithm in an online example scenario

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]/ 9= 0.26		
L_{ov} = (66.67% - 14.81%) = 0.52		
Number of preempted demands = 1 ($\#1 : 15_{2,3}(3)$)		

as load balancing and link overload will be compared with the latest online algorithm such as Smart Alloc from Bahnasse et al. (2018) [Bahnasse (2018)] as described in the subsection 5.4.2.1 considering the online case with different arrival rates (λ). The objective of this scenario is to assess the effect of λ on SKM against the state of art algorithms.

2. Scenario 2: This scenario involved an online simulation under mesh network topology and various generated traffic load (same load in all slices, high load in lower priority slices and high load in higher priority slices) for traffic slices of all priorities as detailed in 5.4.2.2. The objective of this scenario is to assess the impact of mesh topology in which all nodes are reachable in a single hop from each other node on the performance of SKM against the state of the art algorithms considering different load distributions.
3. Scenario 3: This scenario involved an online simulation under NSF network topology and various generated traffic load for traffic slices of all priorities as detailed in 5.4.3. The objective of this scenario is to analyze the impact of NSF network on our proposed algorithm performance against the state of the art algorithms considering different load distributions. This is because NSF topology faces more bottlenecks which further complicates resource allocation and QoS management compared to mesh topology.

5.4.2 Simulation settings and obtained results

This section presents simulation settings, the results obtained from the different scenarios and their analysis.

5.4.2.1 Scenario 1: Impact of arrival rate

In this scenario, we assess the impact of different arrival rates on the performance of SKM strategy against other sates of the art algorithms in the network topology adopted in [Bahnasse (2018)], with the aim of optimizing resource utilization while improving acceptance rate in higher priority slices.

Table 5.4: Comparing SKM to Smart Alloc, AllocTC, RDM and MAM algorithms

Item	SKM	Smart Alloc	AllocTC	RDM	MAM
Scenario	Online				
Goal	Maximize overall resource utilization				
Strategy	Confirm slices' constraints. Check available links capacities then allocate				
Resource allocation	The squatting strategy allows sharing unused resources between all $CTs(l)$	It is classified demands based on their threshold. Whatever the priority of the resources required belonging to the high threshold, the latter can benefit from the loans of the other slices	It allows an opportunistic sharing of the link resources among the different slices	The lower priority $CTs(l)$ can reuse the free resources of higher priority $CTs(l)$ and no the reverse	Each $CT_c(l)$ has its private resources, and if the latter is not used, it cannot be allocated to another $CT_c(l)$
Best path selection	Select the highest available resource path. If two or more paths have the same resources, determine which resource is the least consumed path.				
P_{HTL}	Yes	Yes	Yes	Yes	No
P_{LTH}	No	No	Yes	No	No
$K_q(l)$	Yes	No	No	No	No
E2E delay	Yes				
Note	All simulations 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.				

We used in this scenario a substrate network with 8 nodes and 9 links, the link bandwidth resources are given as real numbers, chosen as 150 or 300 units, and the delay in each substrate link was set to 1 ms. The number of slices per link is equal to 3 slices in links that contain 150 units have the same capacity and are equal to 50 units, and slices in links that contain 300 units have the same capacity and equal to 100 units. We assume that the demands arrive with an exponentially distributed lifetime with an average of 100 time units. In this evaluation scenario, the choice of both source and destination nodes for each request is randomly determined. The arrival rate of incoming demands λ is varied from 1 to 4 per 100 time units, over simulation time of 20,000 units. The size of demands were real numbers uniformly distributed between 1–20 units, while the delay for each demand was randomly selected between 1 and 5. For the routing step, using the k-shortest path, the maximum value of k was set to 5. Table 5.5 summarizes all simulation scenarios parameters.

From the results in Fig. 5.4a and Fig. 5.4c, the average links utilization and acceptance ratio for SKM, Smart Alloc and AllocTC resulted in 78.5% for U and 59.70% for AR, which were higher than MAM and RDM by 5.09% and 3.2% for U and by 3.98% and 2.3% for AR respectively. As expected, SKM outperforms the rest of the algorithms in terms of average U_3 and average AR_3 by 11% and 8% respectively, for the different arrival rates (see Fig. 5.4a and Fig. 5.4c). In case of including E2E delay, SKM links' utilization is less than without delay, since considering delay constraints leads to lower AR, resulting in low resource utilization, thus has less utilized substrate links (see Fig. 5.4b

Table 5.5: Simulation scenarios Parameters

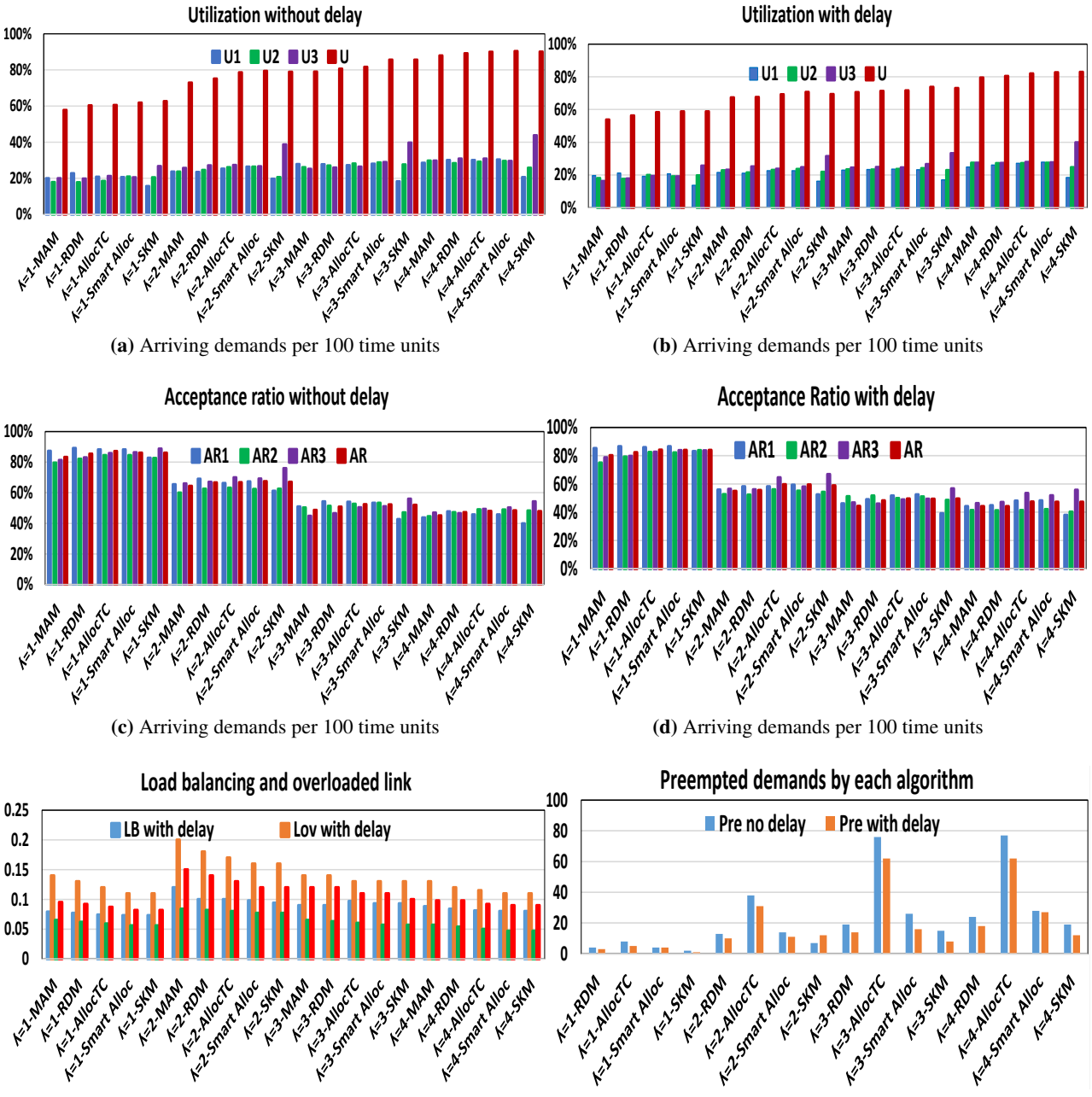
Substrate Network			
Parameter	**Value**		
	scenario 1	**scenario 2**	**scenario 03**
Nodes/Links	8/9	Mesh 5/10	NSF 14/21
Link delay	1 ms	1 ms	1 ms
Link capacity	fixed.[150 units or 300 units]	150 units	150 units
Traffic slices capacity	fixed,[50 units or 100 units]	50 units	50 units
Traffic slices priorities	Unif,[1 - 3]	Unif,[1 - 3]	Unif,[1 - 3]
Time units	0 - 20,000	0 - 25,000	0 - 40,000
Generation rate	Unif,[1 demand - 4 demand]/100 unit times	2500/1 unit time	4000/1 unit time
Demands/lifetime	500 time units	1 time unit	1 time unit
K value for K shortest path	5	5	10
Demands			
	scenario 1	**scenario 2**	**scenario 3**
Source	Random	Random	Random
Destination	Random	Random	Random
Demands/size	Unif, [1 unit - 20 units]	1 unit	1 unit
Demands/lifetime	500 time units	1 time unit	1 time unit
Delay	Unif,[1 ms - 5 ms]	Unif,[1 ms - 5 ms]	Unif, [1 ms - 10 ms]
Load volume traffic for each slice per-each unit time in each scenario experiments	-	Experiment 1, high load in lower priority slices: 1250, 833, 417 Experiment 2, same load in all priority slices: 833, 833, 833 Experiment 3, high load in higher priority slices: 417, 834, 1250	Experiment 1, high load in lower priority slices: 2000, 1500, 500 Experiment 2, same load in all priority slices: 1333, 1333, 1334 Experiment 3, high load in higher priority slices: 500, 1500, 2000

and Fig. 5.4d).

Fig. 5.4e results show that SKM, AllocTC and Smart Alloc have very close performance with and without delay when increased load in terms of average LB and average L_{ov} due to the algorithms have similar utilization. On the other hand, MAM and RDM showed lowest performance with and without delay among other algorithms in terms of LB and L_{ov} by 0.011, 0.0027 for LB and by 0.025, 0.015 for L_{ov} respectively, where more links not being fully used across the network. In the case of including E2E delay, SKM links' load balancing performance is less than without delay, mainly since SKM with E2E delay utilized less network resources, thus has more overloaded substrate links.

As shown in Fig. 5.4f, SKM outperforms RDM, Smart Alloc and AllocTC in terms of average P_{re} by 5, 10 and 23 demands respectively due to kicking operation. In the case of including E2E delay, number of preempted demands of SKM is less than without delay, mainly since SKM with E2E delay accepted less number of demands.

Impact of delay on all algorithms: The impact of E2E delay on all algorithms, was negative in general overall simulations as shown in Fig. 5.4. The results reflect how the algorithms are performing better without delay than when they were included.



(a) Arriving demands per 100 time units

(b) Arriving demands per 100 time units

(c) Arriving demands per 100 time units

(d) Arriving demands per 100 time units

(e) Arriving demands per 100 time units

(f) Arriving demands per 100 time units

Figure 5.4: Overall Performance of SKM with and without E2E delay compared to Smart Alloc, AllocTC, RDM and MAM in scenario 1

5.4.2.2 Scenario 2: Performance considering mesh topology

In this scenario, we assess the impact of mesh topology on the performance of SKM strategy with and without delay against MAM, RDM and AllocTC under various traffic loads. Note that the demands are generated in this scenario with a fixed demands lifetime equal to 1-time unit 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 2500 demand. The total number of demands among slices generated until 10 unit time is 25,000 for each experiment (see Table 5.5). Moreover, we consider in this scenario three experiments in order to analyze the performance of SKM under several metrics and different load distributions between different priority slices. Please note that in all experiments, the capacity of each slice along the whole network is 500 unit ($RC_c(l)$ * 10 links = total size of the slice across the network). The evaluation 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.

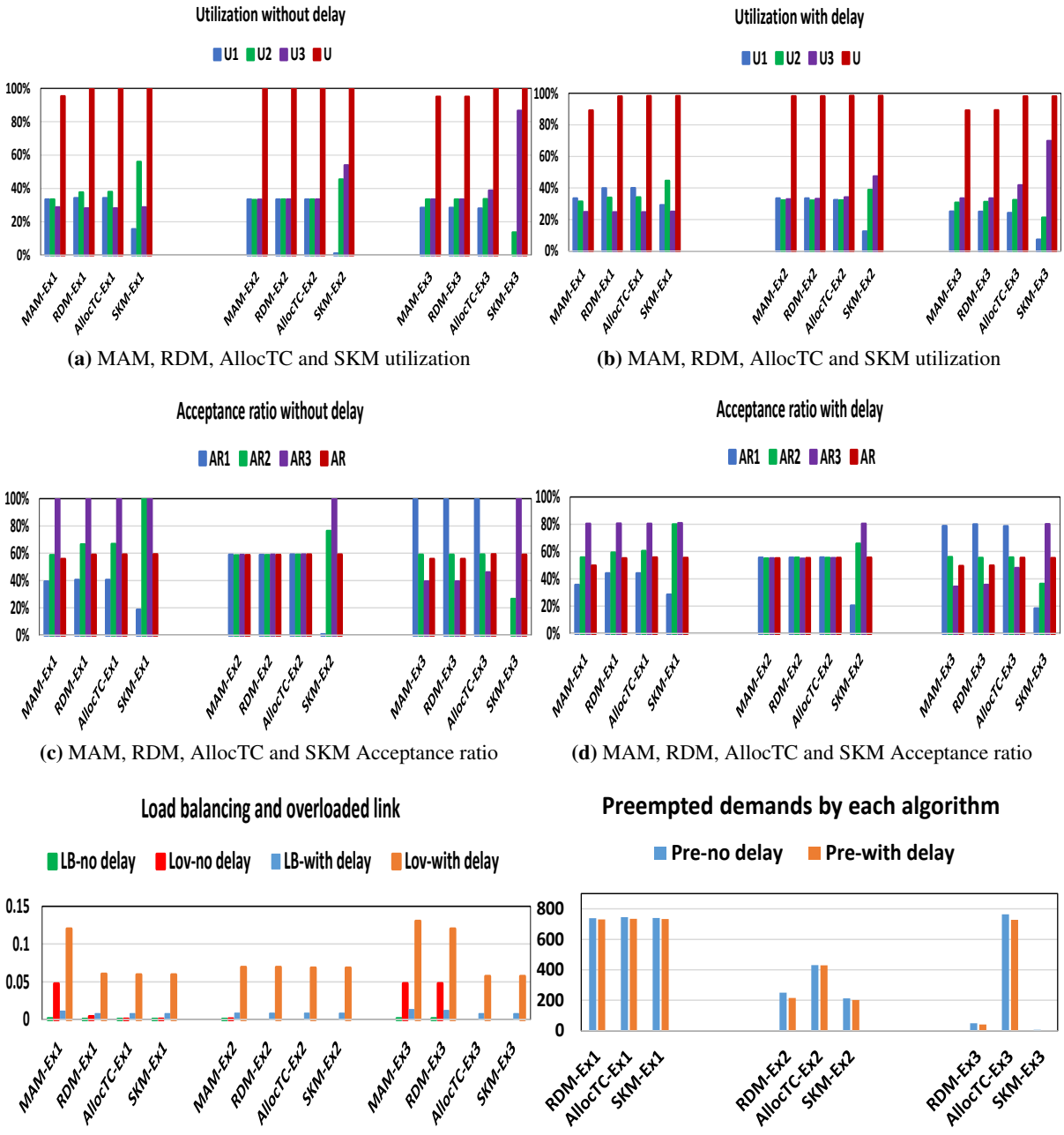
The objective of experiment one is to illustrate that SKM has similar behaviour to RDM and AllocTC at high loads for lower priority slices across the network. The simulation experiment enforces the share or squatting strategy that is inherent to RDM across the network. The objective of experiment two is to illustrate that the SKM guarantees to accept more demands for higher priority slices than AllocTC, RDM and MAM in case of same loads in traffic slices across the network. The objective of experiment three is to illustrate that SKM has an similar behaviour to AllocTC before the saturation case when the load is high for higher priority slices across the network. This is verified by enforcing the share strategy of AllocTC or squatting strategy. Also, SKM achieves more accepted demands than AllocTC and RDM at high loads for higher priority slices, which is due to being stricter on priorities than the other algorithms after saturation case.

Fig. 5.5 shows the results for each algorithm with and without delay in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov} using different traffic load according to experiments 1–3.

Experiment 1, considering high load in lower priority slices: In terms of U and AR, Fig. 5.5a and Fig. 5.5c show that SKM, AllocTC and RDM resulted in 100% U and 59% 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. As expected, SKM, AllocTC, RDM and MAM have similar behaviour in terms of U3 and AR3 by achieving 25.60% and 100% (417/417) AR3, respectively. This is due to the fact that 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 MAM by 18.02%, 18.34%, 22.54% in terms of U2 due to kicking operation. The same trend of performance is observed in Fig. 5.5c in terms of AR_c.

Fig. 5.5e illustrates that SKM, AllocTC and RDM have a very close performance in terms of LB and L_{ov}, this is because these algorithms have a similar value of network resource utilization which is 100% U (almost all links are fully used). Moreover, MAM gives the lowest performance in terms of LB and L_{ov} resulted in 0.0011 LB and 0.047 L_{ov} where more links not being fully used across the network. Moreover, as shown in Fig. 5.5f, SKM, AllocTC and RDM resulted in 740, 739 and 745 in terms of P_{re} respectively.

Experiment 2, considering same load in all priority slices: Fig. 5.5a and Fig. 5.5c illustrate that



(a) MAM, RDM, AllocTC and SKM utilization

(b) MAM, RDM, AllocTC and SKM utilization

(c) MAM, RDM, AllocTC and SKM Acceptance ratio

(d) MAM, RDM, AllocTC and SKM Acceptance ratio

(e) Load balancing and overloaded link by each algorithm

(f) RDM, AllocTC and SKM number of preempted demands

Figure 5.5: SKM with and without E2E delay performance compared to MAM, RDM and AllocTC in scenario 2

the SKM, AllocTC, RDM and MAM resulted in 100% U and 58.88% AR where 1472 demand from 2500 are accepted per each unit time. Moreover, SKM outperforms MAM, RDM and AllocTC in the highest priority slice by 20.47% in terms of U3 and 41.17% in terms of AR3. Also, from the results, 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 behaviours) due to the kicking operation.

Figs. 5.5e 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. Moreover, as shown in Fig. 5.5f, SKM outperforms AllocTC and RDM by 219 and 38 in terms of P_{re}, respectively due to kicking operation.

Experiment 3, considering high load in higher priority slices: Fig. 5.5a and Fig. 5.5c illustrate that the 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 are 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. Furthermore, SKM outperforms AllocTC, RDM and MAM in the highest priority slice by 47.79%, 53.14%, 53.14% in terms of U3 and by 54.28%, 60.95%, 60.95% respectively in terms of AR3.

Fig. 5.5e illustrate that SKM and AllocTC have a similarly good performance by achieving zero in terms of both LB and L_{ov} since all links are fully used in the network. Moreover, RDM and MAM gave the worst performance in terms of LB and L_{ov} and resulted in 0.0117 and 0.0474, respectively, where more number of links are not fully used across the network. Further, from the results of Fig. 5.5f, 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, no need to use the kicking operation.

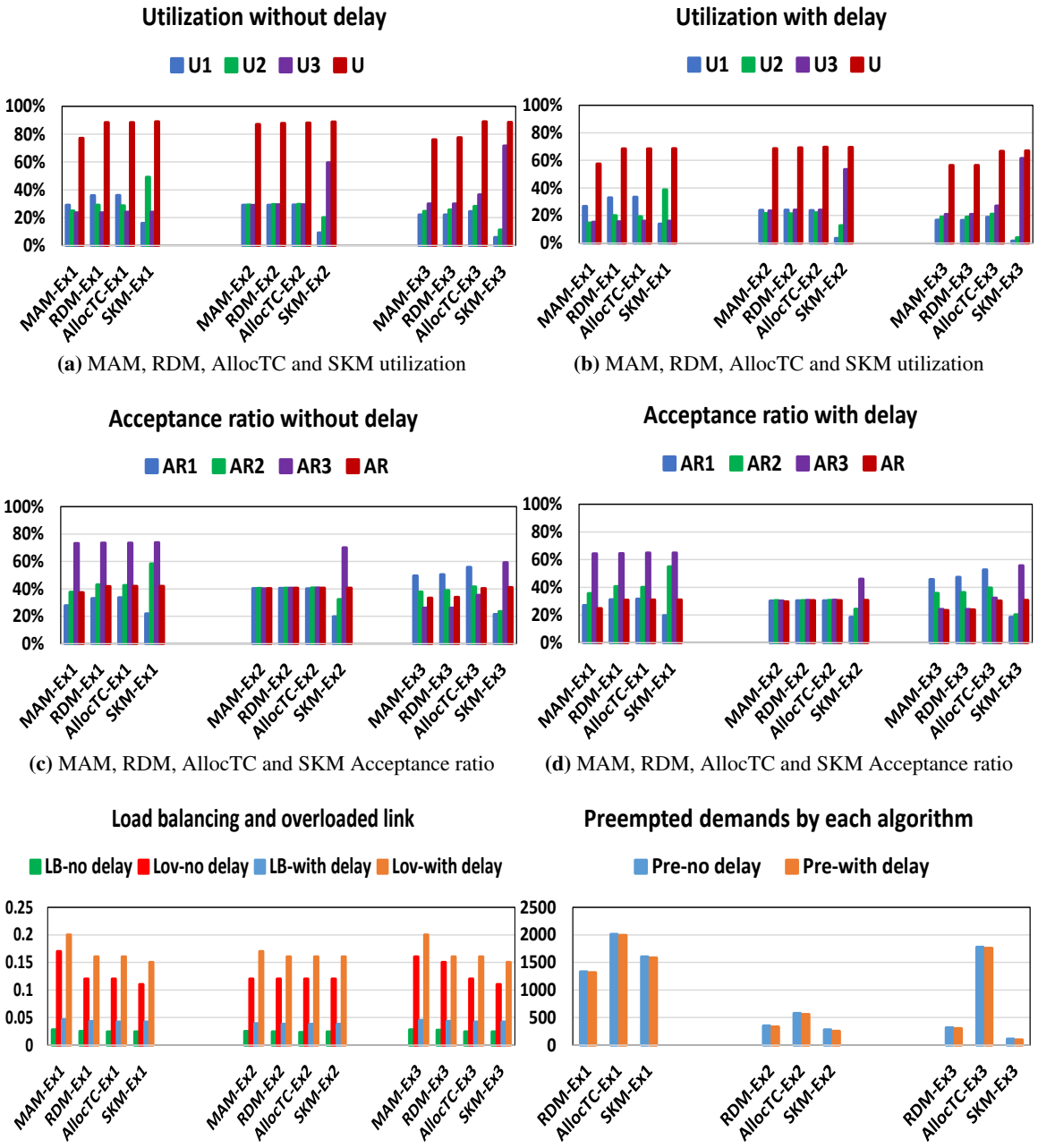
Impact of delay on the performance of different algorithms: All simulation results showed that impact of delay on resource allocation process was clearly the most significant parameter among all varied metrics while testing SKM. Specifically, referring to Fig. 5.5b and Fig. 5.5d, SKM's average utilization and average acceptance ratio with delay, were less than when it was not included by 3%, and 5%, respectively. Similar trends can be seen by referring to SKM's results for average U_c, AR_c, LB, L_{ov} and P_{re} (see Fig. 5.5e and Fig. 5.5f).

5.4.3 Scenario 3: Performance considering NSF topology

In this scenario, we assess the impact of NSF topology on the SKM performance against MAM, RDM and AllocTC under different traffic loads and under fixed demands lifetime, in terms of U, AR, U_c, AR_c, P_{re}, LB and L_{ov}. Moreover, in this scenario we used the same experiments that were considered in the second scenario. Please note that in all experiments, the capacity of each slice along the network is 1050 unit ($RC_c(l)$ * 21 links = total size of the slice across the network).

Fig. 5.6 shows the results by 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.

Experiment 1, considering high load in lower priority slices: From Fig. 5.6a and Fig. 5.6c results, SKM, AllocTC and RDM resulted 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 achieved the lowest performance and resulted in 77.24% U and 37.30% AR where 1492 demands are accepted from 4000 demands per each unit time. Moreover, SKM, AllocTC, RDM and MAM have similar behaviour in terms of both U3 and AR3 by achieving 23.98% and 73.72%, respectively because the load distributions on slice 3 across the network was lower than its capacity. Furthermore, SKM outperforms AllocTC, RDM and



(a) MAM, RDM, AllocTC and SKM utilization

(b) MAM, RDM, AllocTC and SKM utilization

(c) MAM, RDM, AllocTC and SKM Acceptance ratio

(d) MAM, RDM, AllocTC and SKM Acceptance ratio

(e) Load balancing and overloaded link by each algorithm

(f) RDM, AllocTC and SKM number of preempted demands

Figure 5.6: SKM with and without E2E delay performance compared to MAM, RDM and AllocTC in scenario 3

MAM by 20.14%, 20.61%, 24.32% in terms of U2. Further, in terms of ARc, SKM, achieved 12.5% for slice 2 more than MAM, RDM and AllocTC by 15.29%, 15.71%, 20.63%, respectively due to kicking operation (see Fig. 5.6a and Fig. 5.6c).

Figs. 5.6e illustrate that SKM, AllocTC and RDM have a very close performance in terms of LB and L_{ov} resulting in 0.025 and 0.12 respectively, due to the fact that the algorithms have similar

utilization performance. On the other hand, MAM gives the lowest performance in terms of LB and L_{ov} resulting in 0.028 and 0.17 respectively, where more links are not being fully used across the network. Moreover, Fig. 5.6f reveals that SKM, AllocTC and RDM resulted in 1601, 2012 and 1331 in terms of P_{re} respectively due to kicking and preemption operations as we explained earlier.

Experiment 2, considering same load in all priority slices: From Fig. 5.6a and Fig. 5.6c results, SKM, AllocTC, RDM and MAM resulted 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 same in all slices. Moreover, 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. Furthermore, Fig. 5.6e shows that the performance of SKM, AllocTC, RDM and MAM are similar in terms of LB and L_{ov} resulted in average 0.024 and 0.12, respectively due to the algorithms have similar utilization performance. Further, Fig. 5.6f shows that the SKM outperforms AllocTC and RDM by 298 and 71 in terms of P_{re}, respectively due to kicking operation.

Experiment 3, considering high load in higher priority slices: From Fig. 5.6a and Fig. 5.6c results, SKM and AllocTC have similar performance in terms of U and AR by achieving 88.65% U and 41.05% where 1642 demand are accepted from 4000 demand per each unit time. On the other hand, RDM and MAM performance are 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. From Fig. 5.6a and Fig. 5.6c illustrate that the SKM outperforms AllocTC, RDM and MAM in the highest priority slice by 35.17%, 41.71%, 41.71% in terms of U3 and by 23.8%, 33.19%, 33.19% respectively in terms of AR3 (as the expected from the behaviours) due to kicking operation.

From the results of Fig. 5.6e illustrate that SKM and AllocTC have a similarly good performance by achieving 0.025, 0.12 in terms of both LB and L_{ov}. Moreover, RDM and MAM gave the worst performance in terms of LB and L_{ov} and resulted in 0.028 and 0.16, respectively, where more number of links are not fully used across the network. Moreover, Fig. 5.6f, SKM outperforms RDM and AllocTC by 1665 and 205 respectively in terms of P_{re} since the load was was too low on the lower priority slices so, no need to use kicking operation of SKM.

Impact of delay on the performance of different algorithms: All simulation results showed that impact of delay on resource allocation process was clearly the most significant parameter among all varied metrics while testing SKM performance across NSF network. Specifically, referring to Fig. 5.6b and Fig. 5.6d, SKM's average utilization and average acceptance ratio with delay, were less than when it was not included by 20.42%, and 10.19%, respectively. Similar trends can be seen by referring to SKM's results for average U_c, AR_c, LB, L_{ov} and P_{re} (see Fig. 5.6f and Fig. 5.6e).

These values confirm the importance of including E2E delay as a main constraint when solving resource allocation problem, as a direct evaluation metric for real world 5G networks.

5.4.4 Analysis of simulation results

Through the simulation analysis of all the algorithms in the considered scenarios mentioned in subsections 5.4.2.1, 5.4.2.2 and 5.4.3, the following points can be obtained:

1. Addressing delay problems: Incorporation of E2E delay constraint had a vital impact on the resource allocation process, as displayed by lower resource utilization and acceptance ratios across all simulations in the range of 10% and 4% respectively when compared to the cases without delay.
2. Overall online performance of SKM, measured by resource utilization and acceptance ratio were in average in the range of 78.5% and 59.70% respectively, which is similar to the latest referenced online algorithm by Bahnasse et al., (2018). Moreover, SKM outperforms all algorithms in terms of average U_3 and average AR_3 by 11% and 8% respectively, for the different arrival rates (see Fig. 5.4a and Fig. 5.4c).
3. The impact of the topology on the SKM performance and other algorithms is discussed below: Our analysis shows that the all algorithms achieve worse performance in terms of links utilization in NSF topology compared to Mesh, even though NSF have more nodes and links. This is because the mesh topology exhibits a low betweenness centrality value compared to NSF, as a result, the mesh topology experiences fewer bottlenecks compared to NSF topology. This is due to the fact that under mesh topology, all nodes are reachable in a single hop from each other, hence, bottlenecks are minimal since most demands are mapped on single edge paths. In addition, the mesh topology has a high closeness implying on average, mapping of demands from source to destination uses fewer links (shortest path length). All the above issues account for the better performance in terms of AR, U, load balancing, delay, resource consumption and number of preempted demands among others for the mesh topology across all the algorithms. In online experiment 2 (same load in all slices), the performance improvement of all algorithms such as SKM under mesh network in terms of U and AR is 9.48% for U and 14.69% for AR compared to NSF. Note that, we found similar attribute for the other experiments.
4. Usage recommendations: under online scenario: SKM is a suitable algorithm to be used under different topologies but from our experiments we found that SKM provides high performance in terms of AR, U, LB, Lov and P_{re} under topolgies with fewer bottlenecks such as mesh topology irrespective of the load distributions. Moreover, SKM performance gain was more significant with high load in higher priority slices compared to other strategies in terms of accepting higher priority demands. In addition, SKM can reproduce the behaviour of MAM, RDM, and AllocTC in a single model and, as such, generalizes the inherent behaviour of these BAMs in a single implementation in case of unsaturated network.
5. Execution time: We investigated the impact of processing and time costs. The proposed algorithm performance has a sorting step, which needs slightly more memory, but we did not calculate and focus on the cost in terms of memory because our focus was the run time of the algorithms. For example, when the E2E delay was not included, SKM achieved 10 h, 7 min and 56 s as average run-time to assign the demands after running the algorithms 10 times using experiment 3 of scenario 3. RDM and AllocTC have a slightly lower run time complexity (60 and 20 min respectively) than SKM. Nevertheless, SKM gave very high utilization and acceptance ratio in higher priority slices. Also, when we compare the proposed algorithm

with MAM, SKM's run time complexity is approximately 1 h and 45 min more than MAM. Incorporating E2E delay constraints is expected to increase run time due to additional need to search for more paths, however, since we adopted the K shortest path approach, those additional paths are few in number, the additional run time was insignificant compared to the case without delay constraint.

5.5 Conclusions

This chapter introduced a new algorithm based on SKM that efficiently allocates, manages and controls the slice resources under several constraints for 5G and beyond networks, such as priority, bandwidth and E2E delay in real-time while aiming to maximize the overall resource utilization in the substrate network.

In a practical scenario, the computational needs suggest to be executed inside NFV to provide intelligent decisions regarding admission control, routing path computation and resource allocation with a goal of dynamic resource management and guarantee QoS constraint routing for intelligent network slicing management. Moreover, the algorithm proposed is stricter on priorities and significantly differentiates priorities, especially under congested scenarios to optimize usage and provide high acceptance for users of the higher traffic priority slice, which is critical to ensuring the quality of service.

The experimental results showed that the best available algorithm to handle slices until now, SKM without delay constraint managed to maximize the average resource utilization in the substrate multi-hop network by 20.42% and by 3% in the substrate single-hop network. Additionally, this algorithm achieved up to 100% acceptance ratio in higher priority user slices which can not be achieved by other algorithms in some scenarios. However, when the E2E delay constraint is considered, SKM performance is degraded across all evaluation metrics, suggesting that, introducing E2E delay as the main constraint had a clear impact on the whole resource allocation process, and so, it has to be one of the key metrics when evaluating real-world 5G networks.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

CHAPTER 6

Conclusions and Future Work

This chapter will present the key results from the proposed algorithm in this study, accompanied by some proposals for potential studies that can exploit those findings. This study presented a service deployment algorithm that uses SKM intelligence to effectively allocate, manage, and control slice resources under several constraints in a multi-slice scenario, such as priority, bandwidth, and E2E delay with targeting to maximize the total resource usage in the substrate network. Moreover, this algorithm can handle service requirements with delay constraints, in off-line and on-line modes. The chapter lists the impacts of SKM on a single link and real 5G networks in various traffic loads, across offline and online scenarios. Regarding potential recommendations, the chapter summarizes some relevant additional studies that can enhance the efficiency and applicability of the proposed algorithms, and then it presents some suggestions about using machine learning algorithms coupled with the SKM methodology, and implement them for better resource allocations.

6.1 Findings about SKM algorithm

The key benefits of Network Slicing arise as one of the big enablers to deal with strict new specifications of 5G networks and beyond. One of the fundamental challenges of Network Slicing is setting up physical network infrastructure and separating them across various slices. 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, in this thesis, we propose an resource allocation strategy for multiple slice networks based on resources utilization optimization using a proposed and analyzed SKM. SKM is a suitable algorithm for dynamically allocating network resources to different priority slices along paths and improving resource utilization under congested scenarios.

6.1.1 SKM for efficient resource allocation in a multi-slice scenario

The SKM techniques was applied in the simulations in a single link to serve demands belonging to slices with different priority in order to prove the concept in **chapter 3** for offline and online scenarios. **Chapter 4** focuses on how intelligence can be deployed in NFV in order to provide efficient utilization of link bandwidth resources based on SKM techniques in a multi-slice scenario considering strong constraints as required in offline and online 5G scenarios. In **chapter 5** we

analyze the impact of delay constraint on the performance of an online resource allocation algorithm based on an intelligent efficient squatting and SKM, proved in **chapter 3 and 4** to be the most effective up to the present time yet. Simulation results showed that thanks to our proposed offline and online SKM algorithms, 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. Consequently, that was translated into 100% resource utilization and acceptance ratio for higher priority slices in scenarios where the other state of art algorithms not able to reach by far in some scenarios. Simulation results showed that incorporating delay constraints has a significant impact on the performance, resulting in up to 10% and 4% reduction in terms of average resource utilization and acceptance ratios respectively.

6.1.2 Offline and online SKM techniques in a single sliced link

In chapter 3, a novel resource management model has been proposed, able to guarantee 100% utilization even when different priorities are considered, permitting the usage of the full resource even when no higher priorities are requesting it. This is a major difference compared to other proposals. The SKM starts out as a simple MAM algorithm, very conservative, but the behaviour changes when it requires more resources, becomes more aggressive when higher priorities cannot obtain enough resources. Simulations has validated the proposal and has shown the performance in a single link in terms of utilization and acceptance ratio, including those metrics per priority class. The simulations results showed that irrespective of the load distribution among classes as shown in Table 3.8 and Figs. 3.11a - 3.11f SKM guaranteed 100% acceptance ratio for the higher priority users (class 4, class 3) whenever the higher priority demand does not exceed the available network resources as compared to 66.67% for RDM and AllocTC respectively. Moreover, SKM's performance model gives the same or very close to FIFO in terms of scalable allocation of resources from either low or high classes and in addition to the feature of providing the QoS by considering the priorities in the link. Finally, SKM is a very interesting strategy in relation to some emerging technologies that feature diverse QoS requirements and priority admission control. This is the case for 5G mobile network scenario which is expected to serve flexible and varied requirements hence there is a need for dynamic allocation of network resources according to the demands. Network slicing scenario, where the different slices have varying priorities in terms of admission and resource allocation. Also in VNE scenario, the need for resource management where physical resources require sufficient reservation plus allocation phases to satisfy virtual demands on top of a substrate network that has limited residual capacities.

6.1.3 SKM online techniques in a multi-slice full network topology

From the results, the proposed service deployment algorithm used SKM techniques resulting in 100% resource Utilization in bandwidth-constrained environments in some scenarios (as shown in Fig. 4.7) that leads to an increment in revenue for network service providers, improves the experience for the user. Irrespective of the load division among slices, the proposed algorithm achieves 100% admission for higher priority users (MIoT slice, uRLLC slice) as compared to the range of 38.54%

for other algorithms (see Fig. 4.7) which meaning that the proposed algorithm guarantees a high level of isolation and reducing SLA violation penalties. Moreover, as shown in Fig. 4.8, the algorithms used SKM and AllocTC strategies 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, the algorithms used RDM and MAM strategies 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. 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.

6.1.4 SKM offline techniques in a multi-slice full network topology

Simulation results as shown in Fig. 4.9c and Fig. 4.9f showed that as demand size increases as in the case where the size is 2001, the proposed algorithm outperforms the algorithms used AllocTC, RDM and MAM strategies 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. 4.10f reveals that as demand size increases as in the case where the size is 2001, the algorithms used SKM and AllocTC strategies 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, the algorithms used MAM and RDM strategies 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, the algorithms used SKM and AllocTC strategies 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 the proposed algorithm and the algorithm used AllocTC strategy where higher slices can share all unused resources from the lower ones while this is not possible in the algorithms used RDM and MAM strategies. Therefore, the algorithms used RDM and MAM strategies had the lowest performance as they result in 89.13% for U and 66.07% for AR.

6.1.5 Comparison between online and offline scenarios

As we illustrated before in chapter four, 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 (see Fig. 4.12g, Fig. 4.12h, Fig. 4.12b and Fig. 4.12e). This is because the resources are unused in the initial stages (unit-times) for the online case.

6.1.6 Network topology impact

From our experiments, in terms of the performance, the topologies with more direct connections (e.g., mesh topology) give better values than other topologies (e.g., NSF topology). However, this is more costly. So, we have a trade-off. The results reveal that all algorithms achieve worse performance in terms of links utilization in NSF topology compared to Mesh, even though NSF has more nodes and links. This is because the mesh topology exhibits a low betweenness centrality value compared to NSF, as a result, the mesh topology experiences fewer bottlenecks compared to NSF topology. This is due to the fact that under mesh topology, all nodes are reachable in a single hop from each other, hence, bottlenecks are minimal since most demands are mapped on single edge paths. In addition, the mesh topology has a high closeness implying on average, mapping of demands from source to destination uses fewer links (shortest path length). All the above issues account for the better performance in terms of AR, U, load balancing, delay, resource consumption and a number of preempted demands among others for the mesh topology across all the algorithms. Regardless of the topology, the proposed algorithm has similar and even superior performance compared to other alternative algorithms in terms of various metrics in some scenarios, making it a suitable candidate for different 5G deployment topologies.

6.1.7 End-to-End delay impact

Overall, as shown in Fig. 5.6, the use of delay in the allocation phase is very important, but also specifically illustrate that, considering delay can cause some significant deterioration in the efficiency of allocation algorithm on the physical network, so essentially due to the physical characteristics and small number of paths with agreed delay. However, for 5G implementations, even these delay findings are not approved and need to be improved big time to meet the minimum acceptance ratios. In the future networks era including 5G, numerous applications are time-delay sensitive, and the importance of low delay in fields such as telemedicine and autonomous driving. However, for various application scenarios, their delay requirements are varied. Therefore, to meet different QoS requirements simultaneously will significantly reducing the utilization of network resources. From the results, incorporation of E2E delay constraint had a significant impact on the resource allocation process, as displayed by lower resource utilization and acceptance rates across all simulations in the range of 10% and 4% respectively when compared to the case without delay which reduce the revenue for network service providers. Also, from the simulations results, regardless of the load distribution between slices, such as in experiment 3 of scenario 2 in chapter 5, when the delay constraints are incorporated, the proposed algorithm and the routing algorithm that used AllocTC strategy have similar performance in terms of U and AR by achieving 68.23% and 30.74% respectively. On the other hand, the performance of the algorithms used MAM and RDM strategies are the lowest one among the four algorithms by achieving 56.94% and 55.58% in terms of U and 23.71% and 22.81% in terms of AR (see Fig. 5.5b and Fig. 5.5d). Moreover, the proposed algorithm outperforms the other routing algorithms by a range of 41.55% in terms of U3 and 34.67% in terms of AR3. Similar trends can be seen by referring to SKM's results for average LB, L_{ov} and P_{re} (see Fig. 5.5e and Fig. 5.5f).

To this end, the proposed algorithm remains a more suitable candidate for delay constrained scenarios such as 5G compared to the other alternative.

6.1.8 Execution time of the proposed algorithm

As we demonstrated before, to meet different QoS requirements simultaneously becomes a challenge. Moreover, the computational complexity to find the best paths from a source node to a destination node is huge. To this end, the emergence of network virtualization technology (e.g., NFV) can overcome this problem. However, it is complex for the existing resource allocation algorithms to meet the requirements of multiple 5G network slices on time delay performance. The results demonstrate that the proposed algorithm is able to process each request in feasible time making it a good candidate for 5G latency sensitive applications. 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. For example, from the considered number of requests of experiment 3 in scenario 3 in chapter 5, the average execution time per admitted request in milliseconds for every accepted request for the various 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 all requests numbers. The proposed algorithm serves each request in milliseconds which means it is a suitable algorithm for 5G delay-sensitive applications.

6.2 Future work - Enhancements

The following subsections detail strategies for changing the SKM techniques and the suggested algorithms. The recommendations cover other places where SKM may be improved without utilizing artificial intelligence techniques.

6.2.1 Future SKM techniques

Throughout this work, network slicing has turned from a management paradigm in 5G networks to a large research subject in the field of computer networks. Choosing optimal routing paths that taking into account stringent QoS requirements via intelligent algorithms and analysis issues are influential as they involve management systems and architectures, business requirements, resource allocation solutions for the data center, the core network, the wireless connectivity, and spectrum management.

The presumption in the SKM algorithm is focused on the principle of fixed physical network pairs in terms of communication, as defined in chapters 3, 4 and 5 in a multi-slice scenario. SKM has the ability to dynamically allocate network resources such as bandwidth, Label Switched Paths (LSP), fiber, slots among others to different user priority slices. Also, SKM can guarantee the correct level of QoS (especially for the higher priority classes) while optimizing the resource utilization across networks. Moreover, given the network slicing scenarios, the proposed scheme can be employed for admission control. However, understanding that potential Internet uses 5G for IoTs and vehicular

terminals among other services is going to be mostly competitive and handheld in nature, another variant of the SKM techniques promoting resource distribution may be built as well.

6.2.2 SKM techniques improvements

The proposed algorithm aims to support users belonging to high priority slices regarding acceptance and resource allocation, thus, better performance of high slices was revealed at the expense of lower priority slices during congested scenarios due to kicking action. It is expected that groups of higher priority applications in multi-service 5G networks could benefit from improved link resource utilization achieved by the proposed algorithm.

However, SKM can be improved by considering aforementioned thresholds to define and guarantee minimum resources for each slice that will avoid resources beat down for lower priority slices. Last but not least, SKM can also be adapted to the allocation of node resources with minimal modification, which we consider as future work.

6.2.3 Delay on 5G/Beyond-5G

In this study, end-to-end delay is established based on the recommendations from [3GPP1 (2020)] and [3GPP2 (2020)], which is linked to improved multiple applications such as mobile broadband and ultra-reliable low latency. However, provided that potential networks are expected to utilize fiber optics in their topologies, then inevitably, they may suffer from the propagation latency, as opposed to other delays including sorting, queuing, and transmission. In future work, SKM strategy may be performed on other forms of networks utilizing wide and dispersed cloud of core and edge datacenters for example, and further tests could be also done about the impacts of the last mile delays by using resource allocation mechanism on a complex network infrastructure.

6.2.4 Solving resource allocation for multiple-hop paths

As future work, the authors are planning to conduct further study in the context of multi-hop paths, considering an E2E delay for specific 5G applications. Moreover, we aim to implement a heuristic to reduce the computational needs and to provide faster response, which is essential in 5G applications.

6.2.5 Resource utilization in mobile edge computing

In 5G, mobile edge computing and small cells are necessary to attain the fast speeds required and would be commonplace. Thus, managing the utilization of the whole resource network is going to be a very difficult job. To maximize resource utilization on the network that use mobile edge computing technologies, future research is required to implement the SKM strategy, adjust the recommended algorithm, and select which lower priority cells could be preempted during congestion case while maintaining service maintainability, coverage, and reasonable signal quality and stability for the cell's users.

In this case, further metrics need to be considered, such as QoS categories for the various apps, data about the edge network setup, including information about the actual resource allocations and utilizations at the edge cloud, as well as effective and fast mobile edge topology exploration, route configuration, handover and other scalability challenges. In addition, the algorithm can employ statistical analysis and background of the edge cloud to allow for effective consolidations and aid choosing the right resources for the new allocations that will maximize overall resource usage and maintain consistency at the edge cloud.

6.2.6 SKM for large scale networks

Future research could investigate designing distributed online resource allocation algorithm, taking into account SKM strategy for managing huge traffic from masses of users in vast networks that involve cloud and edge computing technologies. It would be important to consider other goal roles rather than the utilization maximization aim which relies on servers utilizations, and use other goals and constraints such as, targeting revenue maximization or cost minimizations, assisted by constraints for priority, allocations on various paths, using edge datacenters, in addition to delay and bandwidth.

6.2.7 Solving resource allocation for NFV architecture

As a general application of future infrastructure of the internet, we will attempt to develop a general heuristic model by applying the (SKM) technique to solve the problem of allocating resources in the virtual infrastructure (IaaS) using NFV architecture for online case. Moreover, we aim to develop an algorithm to overcome the complexity of both VNFs scheduling (delayed VNFs processing) and resource allocation problems in NFV environment.

6.2.8 Implementation in a real field experiment

Another part of the research that was significant, but was outside the reach of this study, is to incorporate the slicing process in hardware. The suggested approach would have considerable operational utility if proposed methods could be evaluated in an actual implementation scenario. More tests would be helpful in that they would explicitly consider different APs, customers and slices. The development phase for our proposal would not be difficult since it will be focused on an already deployed hardware framework. In the other side, the implementation of QoS Slicing may be challenging to perform, and more complex. In order to do so, we must combine the scale of the queues and the usable bandwidth in the hardware drivers.

6.3 Future work - Machine learning for 5G/B5G networks

Future 5G/B5G networks should be rapid and capable of dynamic resource allocation for massive connectivity, ultra-low latency, and ultra-high reliability and capacity. Research in artificial intelligence that really can adapt to the changes occurring in 5G networks and the environment and develop the previous experience to improve future system performance can be useful to 5G/B5G systems.

Machine learning methods have been used in a broad range of networking contexts including routing, load balancing, QoS and queuing, admission management, and resource allocation among others.

The following subsections describe some of the study recommendations where the work from this thesis could be improved using some algorithms from the machine learning fields.

6.3.1 Paths optimized through machine learning

Optimization of routing in network service virtualization is a central issue. In 5G networks, the controller may change traffic flows by altering network switches, which monitor routing of data. The controller may guide or steer which traffic flow is to be passed or redirected on a particular course. Inefficient routing choices, which may contribute to overloading of the network connections, would reduce the network's total resource usage and have a detrimental impact on the overall efficiency of the network. Thus, designing the best pathways and optimally routing of traffic flows may be a future research issue, which could exploit the usage of the SKM methodology in this thesis, and add any machine learning methods. The use of a machine-learning methodology to predicting the shortest paths using a SKM strategy would have considerable benefit in this study.

6.3.2 Load balancing with machine learning

The rapid rise in data usage and requests for fast data speeds on new 5G networks is a major problem confronting resource allocation in datacenter management. For e.g detecting traffic congestions in servers and links, calculating pathways bottlenecks, coping with servers and links failures in emergency situations, handling ultra low latency and very sensitive networks for end-to-end delays, and monitoring the resource usage of heavily used servers, all of these can place a substantial computing overheads on datacenters, before they redistribute their loads in a cost efficient way, and on real time bases in fractions of microseconds to comply with 5G specifications.

Accordingly, research can be conducted using the SKM strategy discussed in this thesis, combined with machine learning to design a new algorithm that considers load balancing mechanisms to increase overall network capacity, resources utilization and user throughput while providing faster response time, lower costs, high network reliability and scalability, and efficient resource consumption. Some examples of machine learning algorithms that are relevant to load balancing in datacenters such as neural network and reinforcement learning algorithms, may be combined to build a centralized dynamic load balancing mechanism that considers connection utilizations, end-to-end delay, and overall resource consumption.

6.3.3 QoS with machine learning

QoS metrics such as queue occupancy, packet loss, latency, and throughput are network indicators that are used to monitor and direct the resource allocation phase, and to determine the overall network efficiency. In order to provide higher quality traffic management, the proposed algorithm in this study would use machine learning for traffic prediction and classifications purposes, and an updated version

of the online algorithm will be built utilizing one or a combination of machine learning techniques for quality dependent traffic control.

The updated algorithm can for example forecast traffic congestion and other issues in advance, and can compensate for mistakes during the resource allocation phase, or identify in which way to guarantee the standard of service. Moreover, the updated algorithm may identify the traffic on a node, connection, or the entire datacenter and appropriately segregate network services' flows so that they can be categorized into quality of service grades, or compact their flow entries with some quality of service assurances for more effective control of the resource allocation phase.

Publications and main contributions

Journals

1. Ahmed El-mekkawi, Xavier Hesselbach, and Jose Ramon Piney, "Evaluating the impact of delay constraints in Network Services for Intelligent Network Slicing based on SKM Model." Journal of communications and networks (**Impact Factor: 2.43, Q 2**), 2020. (SUBMITTED 2th-Dec-2020, first round revision was done, and expecting final decision during the writing of this thesis)

   a- Impacts of end-to-end delay constraint on the performance of the suggested online algorithm, SKM, were deeply analyzed, representing direct application under network slicing scenario.

   b- To maximize the overall resource utilization in the whole physical network, SKM was proposed to exploit resources partition and reservation according to different priority slices with the flexibility of using the full amount of resources when they are not demanded by other slice types.

2. Ahmed El-mekkawi, Xavier Hesselbach, and Jose Ramon Piney, "Novel NFV Aware Network Service for Intelligent Network Slicing Based on Squatting-Kicking Model" Journal of IEEE ACCESS (**Impact Factor: 3.745, Q 1**), 2020. [El-mekkawi2 (2020)].

   a- This paper proposes a paradigm based on NFV architecture to provide the massive computational capacity required in the NSs and support the resource allocation strategy proposed for multiple slice networks based on resources utilization optimization using a proposed and analyzed SKM. SKM is a suitable algorithm for dynamically allocating network resources to different priority slices along paths and improving resource utilization under congested scenarios.

   b- The algorithm used a new path construction methodology that facilitates allocating precise resources for different priority slices along the links across the physical network.

3. Ahmed El-mekkawi, Xavier Hesselbach, and Jose Ramon Piney, "Squatting and kicking model evaluation for prioritized sliced resource management" Journal of Computer Networks (**Impact Factor: 3.11, Q 1**), Elsevier, 2020. [El-mekkawi (2020)].

   a- This paper is formally defining and evaluating a self-provisioned resource management scheme through a smart SKM for a single link as prove of concept.

   b- To the best of our knowledge, this is the first time to provide a solution that effectively guarantees high QoS as SKM for high priority traffic and provisions 100% total resource utilization at the same time.

Conferences

1. Ahmed El-mekkawi, Xavier Hesselbach, and Jose Ramon Piney, "A novel admission control scheme for network slicing based on squatting and kicking strategies," Polytechnic University of Valencia Congress, XIII Jornadas de Ingenieria Telematica - in: 2019 12th International

Conference on Transparent Optical Networks (JITEL), Zaragoza, 2019. [El-mekkawi (2019)].

2. Ahmed El-mekkawi, Xavier Hesselbach, and Jose Ramon Piney, "Network function virtualization aware offline embedding problem using squatting-kicking strategy for elastic optical networks," in: 2018 20th International Conference on Transparent Optical Networks (ICTON), Bucharest, 2018. [El-mekkawi (2018)].

References

[Clemm (2020)] Clemm, A. Zhani, M.F. and Boutaba, R. Network Management 2030: Operations and Control of Network 2030 Services. J Netw Syst Manage 28, 721â750 (2020). https://doi.org/10.1007/s10922-020-09517-0.

[ITU-T (2019)] ITU-T FG-NET2030: New services and capabilities for network 2030: description, technical gap and performance target analysis. FG-NET2030 document NET2030-O-027, (2019).

[A. Karimi (2020)] A. Karimi, et al., "On the Multiplexing of Data and Metadata for Ultra-Reliable Low-Latency Communications in 5G," in IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 12136-12147, Oct. 2020,

[A. Clemm (2020)] A. Clemm, M. T. Vega, H. K. Ravuri, T. Wauters and F. D. Turck, "Toward Truly Immersive Holographic-Type Communication: Challenges and Solutions," in IEEE Communications Magazine, vol. 58, no. 1, pp. 93-99, January 2020, doi: 10.1109/MCOM.001.1900272.

[ITU 'Network 2030' (2018)] ITU 'Network 2030': Initiative to support Emerging Technologies and Innovation looking beyond 5G advances. https://blog.3g4g.co.uk/2018/08/itu-network-2030-initiative-to-support.html

[C. Huang (2020)] C. Huang et al., "Holographic MIMO Surfaces for 6G Wireless Networks: Opportunities, Challenges, and Trends," in IEEE Wireless Communications, vol. 27, no. 5, pp. 118-125, October 2020, doi: 10.1109/MWC.001.1900534.

[M. M. Hussain (2019)] M. M. Hussain, M. S. Alam, and M. M. S. Beg, Feasibility of Fog Computing in smart grid architectures, in Proceedings of 2nd International Conference on Communication, Computing and Networking (C. R. Krishna, M. Dutta, and R. Kumar, eds.), vol. 46, pp. 999 -1010, Singapore: Springer Singapore, 2019.

[P. Bellavista (2017)] P. Bellavista and A. Zanni, Feasibility of Fog Computing deployment based on Docker containerization over RaspberryPi, in Proceedings of the 18th International Conference on Distributed Computing and Networking (ICDCN 17), (Hyderabad, India), pp. 1-10, ACM Press, 2017.

[C. Kuo (2017)] C. Kuo, V. Chang, and C. Lei, A feasibility analysis for edge computing fusion in LPWA IoT environment with SDN structure,â in 2017 International Conference on Engineering and Technology (ICET), pp. 1-6, 2017.

[3GPP1 (2020)] 3GPP. System architecture for the 5G System (5GS). Technical Speci-fication (TS) 23.501, 3rd Generation Partnership Project (3GPP), 2020.Version 16.4.0.

[3GPP2 (2020)] 3GPP. Service requirements for the 5G system. Technical Specification(TS) 22.261, 3rd Generation Partnership Project (3GPP), 2020. Version17.2.0.

[Albreem (2015)] Albreem, Mahmoud A. M.. "5G wireless communication systems: Vision and challenges." 2015 International Conference on Computer, Communications, and Control Tech-

nology (I4CT) (2015): 493-497.

[J. Sachs (2018)] J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair, and K. Kittichokechai, 5G radio network design for ultra-reliable low-latency communication, IEEE Netw., vol. 32, no. 2, pp. 24-31, Mar. 2018.

[FG-NET-2030 (2020)] Network 2030 a blueprint of technology, applications and market drivers towards the year 2030 and beyond. White Paper, https://extra net.itu.int/sites /itu-t/focus group s/net-2030/SiteP ages/White.

[ITU-T FG-NET2030 (2019)] ITU-T FG-NET2030: New services and capabilities for network 2030: description, technical gap and performance target analysis. FG-NET2030 document NET2030-O-027, (2019).

[B. Chang (2019)] B. Chang, L. Zhang, L. Li, G. Zhao, and Z. Chen, Optimizing resource allocation in URLLC for real-time wireless control systems, IEEE Trans. on Veh. Technol., vol. 68, no. 9, pp. 8916-8927, Sep. 2019.

[Xu C (2019)] Xu C, Tao L, Wu H, Ye D, Zhang G. Multiple Constrained Routing Algorithms in Large-Scaled Software Defined Networks, 2019, Feb 27.

[A. M. Medhat (2017)] A. M. Medhat, G. A. Carella, M. Pauls, and T. Magedanz, "Orchestrating scalable service function chains in a NFV environment," in Proc. 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, 2017.

[S. Xiao (2018)] S. Xiao and W. Chen, "Dynamic Allocation of 5G Transport Network Slice Bandwidth Based on LSTM Traffic Prediction," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018, pp. 735-739. DOI:10.1109/ICSESS.2018.8663796.

[C. Song (2018)] C. Song et al., "Machine Learning Enabling Traffic-Aware Dynamic Slicing for 5G Optical Transport Networks," 2018 Conference on Lasers and Electro-Optics (CLEO), San Jose, CA, 2018, pp. 1-2.

[C. Marquez (2018)] C. Marquez et al.: "How Should I Slice My Network?: A Multi-Service Empirical Evaluation of Resource Sharing Efficiency", ACM MOBICOM, 2018.

[C. Marquez (2019)] C. Marquez et al.: "Resource Sharing Efficiency in Network Slicing", IEEE Transactions on Network and Service Management (TNSM), 2019.

[El-mekkawi (2020)] Ahmed El-mekkawi, Xavier Hesselbach, Jose Ramon Piney, "Squatting and kicking model evaluation for prioritized sliced resource management," Computer Networks, Vol. 167, ISSN 1389-1286, 2020, doi:10.1016/j.comnet.2019.107006.

[El-mekkawi2 (2020)] A. El-Mekkawi, X. Hesselbach and J. R. Piney, "Novel NFV Aware Network Service for Intelligent Network Slicing Based on Squatting-Kicking Model," in IEEE Access, vol. 8, pp. 223041-223068, 2020, doi: 10.1109/ACCESS.2020.3044951.

[El-mekkawi (2019)] A. El-mekkawi, X. Hesselbach, J.R. Piney, "A novel admission control scheme for network slicing based on squatting and kicking strategies," in: 2019 12th International Conference on Transparent Optical Networks (JITEL), Zaragoza, pp. 1–8, 2019.

[El-mekkawi (2018)] A. El-mekkawi, X. Hesselbach and J. R. Piney, "Network Function Virtualization Aware Offline Embedding Problem Using Squatting-Kicking Strategy for Elastic Optical

Networks," 2018 20th International Conference on Transparent Optical Networks (ICTON), Bucharest, pp. 1-10, 2018, doi: 10.1109/ICTON.2018.8473869.

[ITUSDN3300 (2014)] ITU-T, "Framework of software-define networking", Recommendation ITU-T Y.3300. 2014. https://www.itu.int/rec/T-REC-Y.3300/en.

[ITU-T T-REC-Y.3101 (2018)] T-REC-Y.3101, "Y.3101: Requirements of the IMT-2020 network," https://www.itu.int/rec/T-REC-Y.3101/en.

[Kunz (2019)] Kunz, W.H., Heinonen, K. and Lemmink, J.G.A.M. (2019), "Future service technologies: is service research on track with business reality?", Journal of Services Marketing, Vol. 33 No. 4, pp. 479-487. https://doi.org/10.1108/JSM-01-2019-0039..

[L. U. Khan (2020)] L. U. Khan, I. Yaqoob, M. Imran, Z. Han and C. S. Hong, "6G Wireless Systems: A Vision, Architectural Elements, and Future Directions," in IEEE Access, vol. 8, pp. 147029-147044, 2020, doi: 10.1109/ACCESS.2020.3015289.

[Sharma (2019)] Sharma and X. Wang, Towards massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions, IEEE Communications Surveys Tutorials, pp. 1-1, 2019.

[Cisco (2018)] Cisco, Cisco Global Cloud index: Forecast and methodology 2016-2021, Tech. Rep. C11- 738085-02, Cisco, San Jose, CA, USA, 2018.

[W. Xiang (2017)] W. Xiang, K. Zheng, and X. Shen, 5G Mobile Communications. Springer International Publishing, 2017.

[ClemmJNSM20 (2020)] Clemm, A., Zhani, M.F. and Boutaba, R. Network Management 2030: Operations and Control of Network 2030 Services. J Netw Syst Manage 28, 721–750 (2020). https://doi.org/10.1007/s10922-020-09517-0.

[O. Brien (2019)] O. Brien, C: Why 6G research is starting before we have 5G. https://venturebeat.com/2019/03/21/6g-research-starting-before-5g/

[Li (2018)] Li, Z., Shariatmadari, H., Singh, B., Uusitalo, M.: 5G URLLC: design challenges and system concepts. 2018 in 15th International Symposium on Wireless Communication Systems (ISWCS), IEEE, August 2018.

[A. Clemm3 (2020)] A. Clemm and T. Eckert, "High-Precision Latency Forwarding over Packet-Programmable Networks," NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2020, pp. 1-8, doi: 10.1109/NOMS47738.2020.9110431.

[H. Amirpour (2020)] H. Amirpour, C. Timmerer and M. Ghanbari, "Towards View-Aware Adaptive Streaming of Holographic Content," 2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW), London, United Kingdom, 2020, pp. 1-6, doi: 10.1109/ICMEW46912.2020.9106055.

[G. Liu (2020)] G. Liu et al., "Vision, requirements and network architecture of 6G mobile network beyond 2030," in China Communications, vol. 17, no. 9, pp. 92-104, Sept. 2020, doi: 10.23919/JCC.2020.09.008.

[Darlis (2017)] Darlis, Denny, Kim, Yong, Cahyadi, Willy, Chung, Yeon-Ho. (2017), Holographic image transmission using blue LED visible light communication system. doi:10.31227/osf.io/pexbv.

[A. Clemm4 (2020)] A. Clemm, M. T. Vega, H. K. Ravuri, T. Wauters and F. D. Turck, "Toward Truly Immersive Holographic-Type Communication: Challenges and Solutions," in IEEE Communications Magazine, vol. 58, no. 1, pp. 93-99, January 2020, doi: 10.1109/MCOM.001.1900272.

[B. Blanco (2017)] B. Blanco et al., Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN, Computer Standards Interfaces, vol. 54, pp. 216-228, 2017.

[3GPP (2019)] 3GPP. System architecture for the 5G System (5GS); Stage 2 (Release 15). Technical Specification (TS) 23.501, 3rd Generation Partnership Project (3GPP), 06 2019. https://portal.3gpp.org/desktopmodules/Specifications/ SpecificationDetails.aspx?specificationId=3144. Version 15.6.0.

[Ericsson (2015)] Ericsson. 5G systems (white paper). Technical Report Uen 284 23-3244, Ericsson, January 2015. http://www.ericsson.com/res/docs/whitepapers/ what-is-a-5g-system.pdf.

[Derakhshani (2015)] Mahsa Derakhshani, Xiaowei Wang, Tho Le-Ngoc, and Alberto Leon-Garcia. Virtualization of multi-cell 802.11 networks: Association and airtime control. arXiv preprint arXiv:1508.03554, 2015.

[Heming Wen (2013)] Heming Wen, Prabhat Kumar Tiwary, and Tho Le-Ngoc. Current trends and perspectives in wireless virtualization. In Mobile and Wireless Networking (MoWNeT), 2013 International Conference on Selected Topics in, pages 62– 67. IEEE, 2013.

[Katsutoshi Kusume (2015)] Kusume, K., Fallgren, M., Queseth, O., Braun, V., Gozalvez-Serrano, D., Korthals, I and Boldi, M. (2015). Deliverable D1. 5. Updated scenarios, requirements and KPIs for 5G mobile and wireless system with recommendations for future investigations. ICT-317669 METIS Project, Public Deliverable ICT-317669-METIS D, 1.

[3GPP (2018)] 3GPP. Service requirements for the 5G system; Stage 1 (Release 15). Technical Specification (TS) 22.261, 3rd Generation Partnership Project (3GPP), 12 2018. https://portal.3gpp.org/desktopmodules/Specifications/ SpecificationDetails.aspx?specificationId=3107. Version 15.7.0.

[Sarkar (2017)] Shamik Sarkar, Christopher Becker, Josh Kunz, Aarushi Sarbhai, Gurupragaash Annasamymani, Sneha Kumar Kasera, and Jacobus Van der Merwe. Enabling wifi in open access networks. In Proceedings of the 4th ACM Workshop on Hot Topics in Wireless, pages 13–17. ACM, 2017.

[Coronado (2018)] Estefanía Coronado, Roberto Riggio, José Villa1ón, and Antonio Garrido. Lasagna: Programming abstractions for end-to-end slicing in software-defined wlans. In 2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM), pages 14–15. IEEE, 2018.

[Jorgensen (2019)] Toke Hoiland-Jorgensen, Per Hurtig, and Anna Brunstrom. Polifi: Airtime policy enforcement for wifi. arXiv preprint arXiv:1902.03439, 2019.

[Javan Erfanian (2015)] Javan Erfanian and Brian Daly. 5G White Paper. White paper, NGMN Alliance, March 2015.

[Ofcom (2014)] UK. Ofcom, The Office of Communications. Infrastructure report 2014. ofcom‘s second full analysis of the uk‘s communications infrastructure. Technical report, Ofcom, The Office of Communications, UK., 2014.

[Zaki (2012)] Yasir Zaki. Future Mobile Communications: LTE Optimization and Mobile Network Virtualization. PhD thesis, Faculty of Physics and Electrical Engineering, University of Bremen, May 2012.

[Graham (2015)] Barry Graham. NFV and SDN answer or question? http://www.cnsmconf.org/2015/files/sdnnfv-keynote.pdf, 11 2015. http://www.cnsm-conf.org/2015/files/sdnnfv-keynote.pdf

[Bhattacharjee (1997)] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura. An architecture for active networking. In: Proc. of the 7th IFIP International Conference on High Performance Networking (HPN-97). 1997, pp. 265-279.

[Kreutz (2015)] Diego Kreutz, Fernando M. V. Ramos, Paulo Jorge Esteves Verassimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-Defined Networking: A Comprehensive Survey. In: Proceedings of the IEEE 103.1 (2015), pp. 14-76.

[ONF (2012)] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, 2012. https://www.opennetworking.org/.

[ONF (2014)] ONF, "OpenFlow-enabled SDN and Network Functions Virtualization", ONF Solution Brief, February 2014.

[S. Agarwal (2013)] S. Agarwal, M. Kodialam, T.V. Lakshman, Traffic engineering in software defined networks, in: Proceedings of the IEEE INFOCOM, 2013, pp. 2211–2219. doi: 10.1109/INFCOM.2013.6567024

[McKeown N (2008)] McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J. OpenFlow: enabling innovation in campus networks ACM SIGCOMM Comput. Commun. Rev., 38 (2) (2008), pp. 69-74, DOI: 10.1145/1355734.1355746.

[M. Dallaglio (2016)] M. Dallaglio, N. Sambo, J. Akhtar, F. Cugini, P. Castoldi, YANG model and NETCONF protocol for control and management of elastic optical networks, in: Proceedings of the IEEE Optical Fiber Communications Conference and Exhibition, 2016, pp. 1–3. https://ieeexplore.ieee.org/document/7537779.

[B. Pfaff (2013)] B. Pfaff, B. Davie, The Open vSwitch Database Management Protocol, RFC 7047, 2013, doi: 10.17487/RFC7047

[M. MacFaden (2003)] M. MacFaden, D. Partain, J. Saperia, W. Tackabury, Configuring Networks and Devices with Simple Network Management Protocol (SNMP), RFC 3512, 2003. doi: 10.17487/RFC3512.

[Heuschkel (2017)] Jens Heuschkel, Michael Stein, Lin Wang, and Max Mahlhuser. Beyond the core: Enabling software-defined control at the network edge. In: Proc. of the 2017 International Conference on Networked Systems (NetSys). 2017, pp. 1-6.

[Bi (2018)] Yuanguo Bi, Guangjie Han, Chuan Lin, Qingxu Deng, Lei Guo, and Fuliang Li. Mobility Support for Fog Computing: An SDN Approach. In: IEEE Communications Magazine 56.5 (2018), pp. 53-59.

[Barakabitze (2020)] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, Andrew Hines, 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges, Computer Networks, Volume 167, 2020, 106984, ISSN 1389-1286,

https://doi.org/10.1016/j.comnet.2019.106984.

[ETSI (2013a)] ETSI GS NFV 002 v1.1.1, "Network Function Virtualisation (NFV); Architectural Framework," 2013. www.etsi.org

[ETSI (2013b)] ETSI GS NFV 004 v1.1.1, "Network Function Virtualisation (NFV); Virtualization Requirements," 2013. www.etsi.org

[ETSI (2013c)] ETSI, "Network Function Virtualisation (NFV); Use Cases," GS NFV 001 v1.1.1, 2013. etsi.org/nfv/.

[ETSI (2014a)] ETSI, "Network Function Virtualisation (NFV); Terminology for Main Concepts in NFV," ETSI GS NFV 003 V1.2.1, 2014. etsi.org/nfv/.

[ETSI (2014b)] [ETSI (2014b)] ETSI, "Network Function Virtualization (NFV); Service Quality Metrics," GS NFV-INF 010 v1.1.1, 2014. etsi.org/nfv/.

[ETSI (2016)] ETSI, "Pre-deployment Testing; Report on Validation of NFV Environments and Services," GS NFV-TST 001 v1.1.11, 2016. etsi.org/nfv/.

[Mohammadkhan (2020)] A. Mohammadkhan, K. K. Ramakrishnan and V. A. Jain, "CleanG–Improving the Architecture and Protocols for Future Cellular Networks With NFV," in IEEE/ACM Transactions on Networking, doi: 10.1109/TNET.2020.3015946.

[Hawilo (2014)] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal. NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC). In: IEEE Network 28.6 (2014), pp. 18-26.

[Abdelwahab (2016)] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati. Network function virtualization in 5G. In: IEEE Communications Magazine 54.4 (2016), pp. 84-91.

[Herrera and Botero (2016)] J. Gil Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," in IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 518-532, 2016. DOI: 10.1109/TNSM.2016.2598420

[Mijumbi (2016)] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," in IEEE Communications Surveys and Tutorials, vol. 18, no. 1, pp. 236-262, 2016. https://ieeexplore.ieee.org/document/7243304/

[Obraczka (2016)] Obraczka, K., Rothenberg, C., and Rostami, A. (2016). SDN, NFV and their role in 5G. In SIGCOM16. Brazil. Ericsson.

[Basilier (2016)] Basilier, H., et al. (2016). A vision of the 5G core: Flexibility for new business opportunities. Ericsson Technology Review, 93, 2–15.

[5GPPP (2017)] 5GPPP. (2017). View on 5G Architecture (Version 2.0). 5G PPP Architecture Working Group.

[M. Jiang (2016)] M. Jiang, M. Condoluci, T. Mahmoodi, "Network Slicing Management and Prioritization in 5G Mobile Systems", Euro. Wireless 2016, pp. 1-6, 2016. https://ieeexplore.ieee.org/document/7499297.

[H. Zhang (2017)] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami and V. C. M. Leung, “Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges,” in IEEE Communications Magazine, vol. 55, no. 8, pp. 138-145, Aug. 2017.

DOI:10.1109/MCOM.2017.1600940.

[C. Suzhi (2019)] C. Suzhi et al., "Space Edge Cloud Enabling Network Slicing for 5G Satellite Network," 2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC), Tangier, Morocco, 2019, pp. 787-792. DOI: 10.1109/IWCMC.2019.8766619.

[J. Li (2019)] J. Li, X. Shen, L. Chen, J. Ou, L. Wosinska and J. Chen, "Delay-aware bandwidth slicing for service migration in mobile backhaul networks," in IEEE/OSA Journal of Optical Communications and Networking, vol. 11, no. 4, pp. B1-B9, April 2019, DOI: 10.1364/JOCN.11.0000B1.

[L. Feng (2020)] 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," in IEEE Access, vol. 8, pp. 34538-34551, 2020, DOI: 10.1109/ACCESS.2020.2974812.

[H. Uzawa (2020)] H. Uzawa et al., "Dynamic bandwidth allocation scheme for network-slicing-based TDM-PON toward the beyond-5G era," in IEEE/OSA Journal of Optical Communications and Networking, vol. 12, no. 2, pp. A135-A143, February 2020, DOI: 10.1364/JOCN.12.00A135.

[T. V. K. Buyakar (2020)] T. V. K. Buyakar, H. Agarwal, B. R. Tamma and A. A. Franklin, "Resource Allocation with Admission Control for GBR and Delay QoS in 5G Network Slices," 2020 International Conference on Communication Systems and NETworkS (COMSNETS), Bengaluru, India, 2020, pp. 213-220, DOI: 10.1109/COMSNETS48256.2020.9027310.

[A. Huang (2020)] A. Huang, Y. Li, Y. Xiao, X. Ge, S. Sun and H. Chao, "Distributed Resource Allocation for Network Slicing of Bandwidth and Computational Resource," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020, pp. 1-6, DOI: 10.1109/ICC40277.2020.9149296.

[RFC 2474] F. Baker, D. L. Black, K. Nichols and S. L. Blake, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” RFC 2474 (Dec. 1998). DOI:10.17487/RFC2474.

[RFC 1633] R. T. Braden, D. D. Clark and S. Shenker, “Integrated Services in the Internet Architecture: an Overview,” RFC 1633 (Jun. 1994). DOI:10.17487/RFC1633.

[RFC 2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, “An Architecture for Differentiated Services,” IETF RFC 2475, December 1998. DOI:10.17487/RFC2475.

[RFC 2460] B. Hinden and D. S. E. Deering, “Internet Protocol, Version 6 (IPv6) Speci cation,” RFC 2460 (Dec. 1998). DOI:10.17487/RFC2460.

[RFC 2702] Jim McManus et al., “Requirements for Traffic Engineering Over MPLS,” RFC 2702 (sep. 1999). DOI:10.17487/RFC2702.

[RFC 3272] Anwar Elwalid, XiPeng Xiao, Indra Widjaja, Angela Chiu and Daniel O. Awduche, “Overview and Principles of Internet Traffic Engineering,” RFC 3272 (may. 2002). DOI:10.17487/RFC3272.

[RFC 3564] F. Le Faucheur and W. Lai, “Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering,” IETF RFC 3564, July 2003. DOI:10.17487/RFC3564.

[Hesselbach (2016)] X. Hesselbach, J. Dantas, J. R. Amazonas, J. Botero and J. Piney, “Management

of resources under priorities in EON using a modified RDM based on the squatting-kicking approach," 2016 18th International Conference on Transparent Optical Networks (ICTON), Trento, 2016, pp. 1-5. DOI:10.1109/ICTON.2016.7550386.

[Sadon (2012)] 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. 38 (6) (2012) 1480-1489. DOI:10.1016/j.compeleceng.2012.05.002.

[Adami (2007)] D. Adami, C. Callegari, S. Giordano, M. Pagano, M. Toninelli, "G-RDM: a new bandwidth constraints model for DS-TE networks," in: Proceedings of the IEEE Global Telecommunications Conference, 2007, pp. 2472-2476. DOI:10.1109/GLOCOM.2007.470.

[Tata (2013)] C. Tata and M. Kadoch, "CAM: Courteous bandwidth constraints allocation model," ICT 2013, Casablanca, 2013, pp. 1-5. DOI:10.1109/ICTEL.2013.6632149.

[Trivisonno (2015)] R. Trivisonno, R. Guerzoni, I. Vaishnavi and A. Frimpong, "Network resource management and QoS in SDN-enabled 5G systems," in: Proceedings of the IEEE Global Communications Conference, 2015, pp. 1-7. DOI:10.1109/GLOCOM.2015.7417376.

[Dantas (2014)] J. Socrates-Dantas, R. Melo Silveira, D. Careglio, J. Roberto Amazonas, J. Sole-Pareta and W.V. Ruggiero, "Novel differentiated service methodology based on constrained allocation of resources for transparent WDM backbone networks," in: Proceedings of the IEEE Brazilian Symposium on Computer Networks and Distributed Systems, 2014, pp. 420-427. DOI:10.1109/SBRC.2014.

[Subhashini (2015)] N. Subhashini and A.B. Therese, "User prioritized constraint free dynamic bandwidth allocation algorithm for EPON networks," Indian J. Sci. Technol. 8 (33) (2015) 1-7. DOI:10.17485/ijst/2015/v8i33/72214.

[Neto (2008)] W. da Costa Pinto Neto and J. S. B. Martins, "A RDM-like bandwidth management algorithm for Traffic Engineering with DiffServ and MPLS support, 2008 International Conference on Telecommunications," St. Petersburg, 2008, pp. 1-6. DOI:10.1109/ICTEL.2008.4652679.

[R.F. Reale (2011)] R.F. Reale, W.daC.P. Neto, J.S.B. Martins, "AllocTC-sharing: A new bandwidth allocation model for DS-TE networks," in: Proceedings of the IEEE Network Operations and Management Symposium, 2011, pp. 1-4. DOI:LANOMS.2011.6102265.

[Dures (2017)] G. M. Duraes, A. C. Fontinele, A. B. Soares, R. F. Reale, R. Bezerra and J. S. B. Martins, "Evaluating the applicability of bandwidth allocation models for EON slot allocation," 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bhubaneswar, 2017, pp. 1-6. DOI:10.1109/ANTS.2017.8384163.

[R.F. Reale (2014)] R.F. Reale, Rafael Freitas, Romildo Martins da Silva Bezerra and Joberto S. B. Martins, "G-BAM: A Generalized Bandwidth Allocation Model for IP/MPLS/DS-TE Networks," CoRR abs/1806.07292 (2014): n. pag. DOI:abs/1806.07292.

[Tomovic (2014)] S. Tomovic, N. Prasad, and I. Radusinovic, "SDN Control Framework for QoS Provisioning," Proc. 2014 22nd Telecommunications Forum Telfor (TELFOR), IEEE, 2014, pp. 111–14.

[Dutra (2017)] D. L. C. Dutra et al., "Ensuring End-to-End Qos Based on Multi-Paths Routing Using SDN Technology," Proc. IEEE Global Commun. Conf., GLOBECOM 2017–2017 IEEE, 2017,

pp. 1–6.

[T. Pan (2017)] T. Pan et al., "Opensched: Programmable Packet Queuing and Scheduling for Centralized QoS Control," Proc. 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), IEEE, 2017 pp. 95–96.

[Oliveira (2018)] A. T. Oliveira et al., "SDN-Based Architecture for Providing QoS to High Performance Distributed Applications," Proc. 2018 IEEE Symposium on Computers and Commun. (ISCC), IEEE, 2018, pp. 00 602–07

[Bahnasse (2018)] A. Ayoub Bahnasse et al., "Novel SDN architecture for smart MPLS Traffic Engineering-DiffServ Aware management," in Future Generation Computer Systems, Volume 87, pp. 115-126, 2018. DOI:10.1016/j.future.2018.04.066.

[Tajiki (2017)] M. M. Tajiki, B. Akbari, and N. Mokari, "Optimal QoSAware Network Reconfiguration in Software Defined Cloud Data Centers," Computer Networks, vol. 120, 2017, pp. 71–86.

[Rezaee (2020)] M. Rezaee and M. H. Y. Moghaddam, "SDN-Based Quality of Service Networking for Wide Area Measurement System," IEEE Trans. Industrial Informatics, vol. 16, no. 5, May 2020, pp. 3018–28.

[Qiu (2019)] C. Qiu et al., "A Novel QoS-Enabled Load Scheduling Algorithm Based on Reinforcement Learning in Software-Defined Energy Internet," Future Generation Computer Systems, vol. 92, 2019, pp. 43–51.

[Venkatesh (2019)] K. Venkatesh et al., "QoS Improvisation of Delay Sensitive Communication Using SDN Based Multipath Routing for Medical Applications," Future Generation Computer Systems, vol. 93, 2019, pp. 256–65.

[Montero (2019)] R. Montero et al., "Supporting QoE/QoS aware End-to-End Network Slicing in Future 5G-Enabled Optical Networks," Metro and Data Center Optical Networks and Short-Reach Links II, vol. 10946, Int'l. Society for Optics and Photonics, 2019, p. 109460F.

[Sgambelluri (2019)] A. Sgambelluri et al., "Orchestrating QoS-Based Connectivity Services in a Multi-Operator Sandbox," J. Optical Commun. and Networking, vol. 11, no. 2, 2019, pp. A196–A208.

[Vincenzi (2017)] M. Vincenzi et al., "Multi-Tenant Slicing for Spectrum Management on the Road to 5G," IEEE Wireless Commun., vol. 24, no. 5, 2017, pp. 118–25.

[Sattar (2019)] D. Sattar and A. Matrawy, "Optimal Slice Allocation in 5G Core Networks," IEEE Networking Lett., vol. 1, no. 2, 2019, pp. 48–51.

[Wu1 (2016)] J. Wu, M. Wang and C. Yuen, "Energy-aware concurrent multipath transfer for real-time video streaming to multihomed terminals," 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, pp. 1-6, 2016, doi: 10.1109/ICC.2016.7511547.

[Wu2 (2016)] J. Wu, C. Yuen, B. Cheng, M. Wang and J. Chen, "Energy-Minimized Multipath Video Transport to Mobile Devices in Heterogeneous Wireless Networks," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 5, pp. 1160-1178, May 2016, doi: 10.1109/JSAC.2016.2551483.

[Quang (2018)] P. T. A. c, K. D. Singh, A. Bradai and A. Benslimane, "QAAV: Quality of Service-

Aware Adaptive Allocation of Virtual Network Functions in Wireless Network," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, pp. 1-6, 2018, doi: 10.1109/ICC.2018.8422757.

[Eramo (2018)] V. Eramo and F. G. Lavacca, "Processing and Bandwidth Resource Allocation in Multi-Provider NFV Cloud Infrastructures interconnected by Elastic Optical Networks," 2018 20th International Conference on Transparent Optical Networks (ICTON), Bucharest, pp. 1-6, 2018, doi: 10.1109/ICTON.2018.8473708.

[Kuo (2018)] T. Kuo, B. Liou, K. C. Lin and M. Tsai, "Deploying Chains of Virtual Network Functions: On the Relation Between Link and Server Usage," in IEEE/ACM Transactions on Networking, vol. 26, no. 4, pp. 1562-1576, Aug. 2018, doi: 10.1109/TNET.2018.2842798.

[Zsa (2001)] J; zsa, Bal; zs G; bor, et al. "An efficient algorithm for global path optimization in MPLS networks." Optimization and Engineering 2 (2001): 321-347. doi:article/10.1023/A:1015318600381.

[Wang (1996)] Zheng Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," in IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, pp. 1228-1234, Sept. 1996. doi: 10.1109/49.536364.

[Kodialam (2000)] M. Kodialam and T. V. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Tel Aviv, Israel, 2000, pp. 884-893 vol.2. doi: 10.1109/INFCOM.2000.832263.

[Yang (2001)] Yi Yang, J. K. Muppala and S. T. Chanson, "Quality of service routing algorithms for bandwidth-delay constrained applications," Proceedings Ninth International Conference on Network Protocols. ICNP 2001, Riverside, CA, USA, 2001, pp. 62-70, doi: 10.1109/ICNP.2001.992885.

[Chen (1999)] Shigang Chen. Routing support for providing guaranteed end-to-end quality-of-service. PhD thesis, University of Illinois at Urbana-Champaign, 1999. doi:book/10.5555/871252.

[Tomovic (2016)] S. Tomovic and I. Radusinovic, "Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks," 2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, 2016, pp. 303-311, doi: 10.1109/NETSOFT.2016.7502426.

[Yen (1971)] Yen, Jin Y. "Finding the K Shortest Loopless Paths in a Network." Management Science, vol. 17, no. 11, 1971, pp. 712–716. doi:org/10.1287/mnsc.17.11.712.

[RFC7679 (2017)] G. Almes, S. Kalidindi, M. Zekauskas, and A. Morton, "A One-Way Delay Metric for IP Performance Metrics (IPPM)," 2016. https://tools.ietf.org/html/rfc7679.

[Fischer (2013)] A. Fischer, J. Botero, M. Beck, H. de Meer and X. Hesselbach, 2013. "Virtual Network Embedding: A Survey," in IEEE Communications Surveys and Tutorials, vol. 15, no. 4, pp. 1888-1906. doi: 10.1109/SURV.2013.013013.00155.

[Zaki (2011)] Yasir c, Liang Zhao, Carmelita Goerg, and Andreas Timm-Giel. LTE mobile network virtualization. Mobile Networks and Applications, 16(4):424–432, Jun 2011. ISSN 1572-8153.

https://doi.org/10.1007/s11036-011-0321-7

[Wang (2013)] Xin c, Prashant Krishnamurthy, and David Tipper. Wireless network virtualization. In Computing, Networking and Communications (ICNC), 2013 International Conference on, pages 818–822. IEEE, 2013.

[ladejo (2017)] S. O. Oladejo and O. E. Falowo, "5G network slicing: A multi-tenancy scenario," 2017 Global Wireless Summit (GWS), Cape Town, 2017, pp. 88-92. DOI: 10.1109/GWS.2017.8300476.

[Lucena (2017)] J. Ordonez Lucena, P. Ameigeiras, D. Lopez, J. Ramos-Munoz, J. Lorca and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," in IEEE Communications Magazine, vol. 55, no. 5, pp. 80-87, May 2017. DOI:10.1109/MCOM.2017.1600935.

[Sciancalepore (2017)] V. Sciancalepore et al., "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA. 2017, pp. 1-9. DOI: 10.1109/INFOCOM.2017.8057230.

[Zafar (2011)] Muhammad Salman Zafar, Junaid Zubairi and Aasia Khanum, "Automated traffic engineering using adaptive inter-class mixing," EURASIP Journal on Wireless Communications and Networking, vol. 2011, no. 1, pp. 49, Aug. 2011. DOI: 10.1186/1687-1499-2011-49.

[Tata (2014)] Chafika Tata and Michel Kadoch, "Efficient Priority Access to the Shared Commercial Radio with Offloading for Public Safety in LTE Heterogeneous Networks," Journal of Computer Networks and Communications, vol. 2014, Article ID 597425, 15 pages, 2014. DOI:10.1155/2014/597425.

[Zhang (2007)] D. Zhang and D. Ionescu, "QoS Performance Analysis in Deployment of DiffServ-aware MPLS Traffic Engineering," Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), Qingdao, 2007, pp. 963-967. DOI:10.1109/SNPD.2007.541.

[Han (2018)] B. Han, J. Lianghai and H. D. Schotten, "Slice as an Evolutionary Service: Genetic Optimization for Inter-Slice Resource Management in 5G Networks," in IEEE Access, vol. 6, pp. 33137-33147, 2018. DOI:10.1109/ACCESS.2018.2846543.

[Oliveira (2004)] J. C. de Oliveira et al., "New preemption policies for DiffServ-aware traffic engineering to minimize rerouting in MPLS networks," in IEEE/ACM Transactions on Networking, vol. 12, no. 4, pp. 733-745, Aug. 2004. DOI:10.1109/TNET.2004.833156.

[Reale (2012)] R. F. Reale, W. da C. P. Neto and J. S. B. Martins, "Routing in DS-TE networks with an opportunistic bandwidth allocation model," 2012 IEEE Symposium on Computers and Communications (ISCC), Cappadocia, 2012, pp. 88-93. DOI:10.1109/ISCC.2012.6249273.

[Wang (2012)] Y. Wang, X. Cao, Q. Hu and Y. Pan, "Towards elastic and fine-granular bandwidth allocation in spectrum-sliced optical networks," in IEEE/OSA Journal of Optical Communications and Networking, vol. 4, no. 11, pp. 906-917, Nov. 2012. DOI: 10.1364/JOCN.4.000906.

[Reale (2014)] R.F. Reale, R.M.S. Bezerra, J.S.B. Martins, "A preliminary evaluation of bandwidth allocation model dynamic switching," Int. J. Comput. Netw. Commun. 6 (3) (2014) 131-143. DOI: 10.5121/ijcnc.2014.6311.

[RFC 4125] W. Lai, F. L. Faucheur, "Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering," RFC 4125 (Jun. 2005). DOI:10.17487/RFC4125.

[RFC 4127] F. Le Faucheur, "Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering," RFC 4127, 2005. DOI:10.17487/RFC4127.

[Veres (2007)] S. Veres and D. Ionescu, "A Performance Model and Measurement Framework for DiffServ Implementations," in IEEE Transactions on Instrumentation and Measurement, vol. 56, no. 4, pp. 1473-1480, Aug. 2007. DOI:10.1109/TIM.2007.900422.

[Liu (2007)] C. Liu, Y. Liu, D. Qian and M. Li, "An Approach of End-to-End DiffServ/MPLS QoS Context Transfer in HMIPv6 Net," Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07), Sedona, AZ, 2007, pp. 245-254. DOI:10.1109/ISADS.2007.11.

[RFC 3246] J. Bennett, S. Davari, D. Stiliadis, W. Courtney, K. Benson, J. Y. L. Boudec, V. Firoiu, D. B. S. Davie and A. Charny, "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246. (Mar. 2003). DOI:10.17487/RFC3246.

[RFC 2705] Andrew Dugan, Scott Pickett, Isaac K. Elliott, Mauricio Arango and Christian Huitema, "Media Gateway Control Protocol (MGCP) Version 1.0, IETF," RFC 2705, oct, 1999. DOI:10.17487/RFC2705.

[RFC 3209] Daniel O. Awduche. et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF, RFC 3209, dec, 2001. DOI:10.17487/RFC3209.

[GeraldAsh (2005)] Gerald Ash, "Max Allocation with Reservation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering & Performance Comparisons," RFC 4216, jun. 2005. DOI:10.17487/RFC4126.

[RFC 4128] Wai Lai, "Bandwidth Constraints Models for Differentiated Services (Diffserv)-aware MPLS Traffic Engineering: Performance Evaluation," RFC 4128, jun. 2005. DOI:10.17487/RFC4128.

[McKeown (1999)] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," in IEEE/ACM Transactions on Networking, vol. 7, no. 2, pp. 188-201, April 1999. DOI:10.1109/90.769767.

[Ma1 (2020)] Y. Ma, W. Liang, J. Wu and Z. Xu, "Throughput Maximization of NFV-Enabled Multicasting in Mobile Edge Cloud Networks," in IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 2, pp. 393-407, 1 Feb. 2020, doi: 10.1109/TPDS.2019.2937524.

[Ma2 (2020)] Y. Ma, W. Liang, J. Li, X. Jia and S. Guo, "Mobility-Aware and Delay-Sensitive Service Provisioning in Mobile Edge-Cloud Networks," in IEEE Transactions on Mobile Computing, 02 July, 2020, doi: 10.1109/TMC.2020.3006507.

[Reyhanian (2020)] N. Reyhanian, H. Farmanbar, S. Mohajer and Z. Luo, "Joint Resource Allocation and Routing for Service Function Chaining with In-Subnetwork Processing," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 4990-4994, doi: 10.1109/ICASSP40776.2020.9054706.

[Li (2017)] X. Li et al., "Network Slicing for 5G: Challenges and Opportunities," in IEEE Internet Computing, vol. 21, no. 5, pp. 20-27, 2017, doi: 10.1109/MIC.2017.3481355.

[Zhou (2016)] X. Zhou, R. Li, T. Chen and H. Zhang, "Network slicing as a service: enabling

enterprises' own software-defined cellular networks," in IEEE Communications Magazine, vol. 54, no. 7, pp. 146-153, July 2016, doi:10.1109/CC.2018.8332001.

[OSM (2019)] "OSM Release FIVE Technical Overview," ETSI, Sophia Antipolis, France, 2019.

[Leconte (2018)] M. Leconte, G. S. Paschos, P. Mertikopoulos and U. C. Kozat, "A Resource Allocation Framework for Network Slicing," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, pp. 2177-2185, 2018, doi: 10.1109/INFOCOM.2018.8486303

[NGMN (2016)] "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

[Reale (2016)] Reale, Rafael Freitas, Romildo Martins da S. Bezerra, and Joberto SB Martins. "Applying autonomy with bandwidth allocation models." International Journal of Communication Systems, pp. 2028-2040, 29.13 (2016), https://doi.org/10.1002/dac.3157.
doi: 10.1109/SBRC.2014.50.

[Chowdhury (2012)] M. Chowdhury, M. R. Rahman and R. Boutaba, "ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping," in IEEE/ACM Transactions on Networking, vol. 20, no. 1, pp. 206-219, Feb. 2012, doi: 10.1109/TNET.2011.2159308.

[Kibalya (2020)] Godfrey Kibalya et al., "A novel dynamic programming inspired algorithm for embedding of virtual networks in future networks," Computer Networks, Volume 179, 107349, ISSN 1389-1286, 2020, https://doi.org/10.1016/j.comnet.2020.107349.

[Xu (2019)] Xu, C. et al. "Multiple Constrained Routing Algorithms in Large-Scaled Software Defined Networks." ArXiv abs/1902.10312 (2019): n. pag. doi:1902.10312. 2019 Feb 27.

[Pateromichelakis (2017)] E. Pateromichelakis, K. Samdanis, Q. Wei and P. Spapis, "Slice-Tailored Joint Path Selection & Scheduling in mm-Wave Small Cell Dense Networks," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6, doi:10.1109/GLOCOM.2017.8254139.

[Feng (2016)] Feng, W.; Li, Y.; Jin, D.; Su, L.; Chen, S. "Millimetre-Wave Backhaul for 5G Networks: Challenges and Solutions." Sensors 2016, 16. 892.doi: 10.3390/s16060892.

[Hejja (2018)] Khaled Hejja, Xavier Hesselbach, "Online power aware coordinated virtual network embedding with 5G delay constraint," Journal of Network and Computer Applications, Volume 124, 2018, Pages 121-136, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2018.10.005.

[Kanavos (2021)] Kanavos, A.; Fragkos, D.; Kaloxylos, A. V2X Communication over Cellular Networks: Capabilities and Challenges. Telecom 2021, 2, 1-26. doi: 10.3390/telecom2010001.

[Brown (2016)] Brown, Gabriel. "Exploring 5G new radio: Use cases capabilities and timeline." Qualcomm White Paper (2016): 1-12.

[Ijaz (2016)] Ijaz, Ayesha, Lei Zhang, Maxime Grau, Abdelrahim Mohamed, Serdar Vural, Atta U. Quddus, Muhammad Ali Imran, Chuan Heng Foh, and Rahim Tafazolli. "Enabling massive IoT in 5G and beyond systems: PHY radio frame design considerations." IEEE Access 4 (2016): 3322-3339.