Squatting and Kicking Model Evaluation for Prioritized Sliced Resource Management

Ahmed El-mekkawi, Xavier Hesselbach, and Jose Ramon Piney

Dept. Network Engineering, Universitat Politecnica de Catalunya, C/ Jordi Girona, 1-3 - Edif.C3 - Campus Nord - 08034 Barcelona - Spain

Abstract

Effective management and allocation of resources remains a challenging paradigm for future large-scale networks such as 5G, especially under a network slicing scenario where the different services will be characterized by differing Quality of Service (QoS) requirements. This makes the task of guaranteeing the QoS levels and maximizing the resource utilization across such networks a complicated task. Moreover, the existing allocation strategies with link sharing tend to suffer from inefficient network resource usage. Therefore, we focused on prioritized sliced resource management in this work and the contributions of this paper can be summarized as formally defining and evaluating a self-provisioned resource management scheme through a smart Squatting and Kicking model (SKM) for multi-class networks. SKM provides the ability to dynamically allocate network resources such as bandwidth, Label Switched Paths (LSP), fiber, slots among others to different user priority classes. 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. Simulation results show that our model achieves 100% resource utilization in bandwidth-constrained environments while guaranteeing higher admission ratio for higher priority classes. From the results, SKM provided 100% acceptance ratio for highest priority class under different input traffic volumes, which, as we articulate, cannot be sufficiently achieved by other existing schemes such as AllocTC-Sharing model due to priority constraints.

Keywords: Resource Management, SKM, Utilization Optimization, Class of Service, Dynamic Resource Allocation.

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 [1]. 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) [2]. 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 [3]. 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 [4–9].

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 [10]. 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 [11, 12]. 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 [12].

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

[*] Corresponding author

Email address: ahmed.mohamed.abdelaty.elmekaw@upc.edu, Xavier.Hesselbach@upc.edu, jpiney@entel.upc.edu (Ahmed El-mekkawi, Xavier Hesselbach, and Jose Ramon Piney)

functional and non-functional requirements [13]. Secondly, how to efficiently guarantee QoS and isolation for high priority users especially in congested scenarios while guaranteeing maximum resource utilization [4, 14].

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 [4, 5, 15, 16]. 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 [17]. Moreover, BAMs can handle resource allocation for any resources such as bandwidth, LSPs, fiber, other [18]. 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) [19], Russian Doll Model (RDM) [20], Generalized RDM (G-RDM) [21], AllocTC-Sharing model (AllocTC) [22], 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 [23]. 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 [24]. Nevertheless, these models need to enhance and support differentiated services together with automated, class-based, networking service provisioning.

In light of that, this paper 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 [25, 26]. Moreover, the proposed model can be applied to DiffServ-aware Multi-Protocol Label Switching (DS-MPLS) networks using their TE capabilities [27].

The main contributions of this article are as follows:

1. SKM is a QoS algorithm for multi-class networks and can be used with any networks such as Elastic Optical Network (EON), wireless network, MPLS and among others. We introduced the mathematical definitions of SKM.
2. SKM provides a new policy for selecting and serving demands, which takes QoS constraints into account for different priorities/classes.
3. We evaluated and analyzed the performance of the proposed SKM against the most referenced algorithms in BAMs such as RDM and AllocTC in terms of link load per class, link load, utilization, and acceptance ratio to reflect the ability to manage multi-class demands in a limited resource network, and the ability to adapt to different input traffic volumes. Moreover, according to the strategies that are usually applied in BAMs, to the best of our knowledge, no solution effectively guarantees as high QoS as SKM for high priority traffic and provisions 100% total resource utilization at the same time.
4. Additional evaluations for the proposed SKM were also introduced, for showing the effect of varying demand lifetime on the performance of each scheme under high traffic loading in the higher priority classes, in terms of utilization and acceptance ratio. Also, we compared our proposed algorithm's performance against the most referenced unconstrained algorithms, i.e., First-In-First-Out (FIFO) in terms of utilization and acceptance ratio metrics.

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 classes under different input traffic volumes (especially in congested scenarios). On the other hand, when the traffic is not congested the SKM behaves similar to MAM, RDM and AllocTC.
3. Adaptability to emerging technologies that are characterized by diverse QoS requirements and prioritized admission control, especially under network slicing scenario.

Practical application scenarios:
The SKM 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 the SKM, 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.

The remainder of this paper is organized as follows: Section 2 provides an overview of the related work in literature. Section 3 shows a set of definitions, as well as introduces the notation used in this article. Section 4 elaborates on the existing resource allocation models. Section 5 presents the proposed model and squatting-kicking concepts. Section 6 describes performance evaluation issues. Section 7 explains the asymptotic behaviour of each algorithm in terms of the acceptance ratio with varying lifetime. Section 8 shows a summary of the findings from the simulations. Finally, section 9 concludes the paper and proposes recommendations for future work.

2. Related works

Effective management and allocation of the dynamic resources need smart models which consist of:

- Developing a new QoS algorithm for large scale networks.
- Optimizing the utilization and more restrict on priorities according to the QoS constraints.

2.1. Related works focusing on 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) [28], Integrated Services (IntServ), Internet Engineering Task Force (IETF) [29] and DiffServ [30] were broadly analyzed and implemented. These models based on the specific use of the octet named traffic class [31].

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 [32, 33].

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 [34]. 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 [32].

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.

2.2. 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 [35].

In [36], 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 [21, 24, 37–40]. 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 [22], proposed a model called AllocTC, which provides sharing of the unusable bandwidth of high bandwidth applications priority with low priority and vice versa. In [18], 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 [23], 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. In [27], 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.

In our previous works, [35, 41, 42], we proposed the concept of SKM in order to give 100% overall network utilization while guaranteeing high admission for higher priority classes for any resources (e.g., EON) as the following: In [35] the authors propose a resources allocation method for EON based on a modified RDM using squatting and kicking strategies in order to prioritize the usage of channels and enhancing the total utilization which can also be used as an admission control. The mechanism is described and compared to existing proposals for a few representative numerical examples, regarding the flexibility in the allocation per class. The strategy regards the EON constraints and assumes the optical spectrum to be partitioned and reserved for several different classes. These partitions are allocated according to the priority of the demands and the strategy proposed. This work extends the concepts in [35] by adding online and offline distinction and giving the algorithms in a more formal framework. Moreover, in this work, the performance of the proposed algorithms is analyzed by not only representative examples but also simulations that vary the system parameters. In [41], an offline resource allocation strategy is proposed for EON embedding to improve the computational capacity. The proposed model considers priority classes and utilizes NFV architecture. The proposed algorithm is described, analyzed, and compared with existing models in terms of flexibility in resource allocation per class and prioritization of channel usage. This work differs from [41] in terms of: i) adding online version of the algorithm, ii) using more formal framework for defining the system, iii) evaluating the performance against more recent alternative algorithms (e.g., AllocTC) under various scenario conditions, iv) adding efficient implementation strategies, and v) adding complexity analysis. In [42], we introduced a new flexible admission control mechanism on a pool of resources based on squatting and kicking techniques (SKM) which can be employed under network slicing scenario. The main difference is that the algorithm used in [42] only checks the capacity of the system (i.e., a pool of resources) to potentially accept the demand without considering the links. Therefore, it only focuses on satisfying the simple demands, where the demands are just admitted according to a pool of resources. On the other hand, this study considers the underlying network and the demand structure is much more complicated compared to the previous study in [42] (i.e., the demands are the paths with a required capacity in a given network). Hence, finding routing paths in a given network and allocating/reserving the resources along the path should be considered, which has an extra complexity compared to the setup of [42]. Additionally, this work differs from the previous study in [42] in terms of: i) using more formal framework for defining the system, ii) evaluating the performance under various scenario conditions, iii) adding efficient implementation strategies, and iv) adding complex and in-depth analysis.

3. Definitions and notation

This section has two purposes: The first one introduces the terminology that will be used along with the document, part of which is based on IETF-RFC documents [20, 32, 43–45]. The second purpose is to describe the notation used along with the model's description, analytical model and evaluation sections in the article.

3.1. 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 [20, 43].

- 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 [14, 32, 44].

- 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 [45].

- 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 [45].
- 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) [20]. 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 [20].

  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 [35, 41].
- Kicking: Act or action of expelling a lower priority class from its allocated resources, either partially or totally. In the context of this paper, 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 [35, 41].

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. Notation

A description of all parameters and decision variables used in this article is provided in Table 1 and Table 2 respectively.

Table 1: Parameters of the Model

Abbreviation	Explanation
RC_c	Resource constraints for class c also equal to maximum reservable resources for class c
CT_c	Class of priority c where $c \in [1, N]$ and CT_N is the highest priority class and CT_1 is the lowest priority class.
R	Maximum reservable resources for all classes together and is equal to link capacity
$d_j(CT_c)$	The amount of resources (size) of demand j belonging to class c where $j \in [1, D]$. Where D is the total number of demands by all classes

Table 2: Variables of the Model

Abbreviation	Explanation
D_t	Total number of demands arriving for current unit time
D_c	Total number of demands by class c
D	Total number of demands by all classes
D_c	Total number of demands by class c
D_{c_t}	Total number of demands arriving for each class for each unit time
S_c	The actually allocated resources to class c
BD	Number of blocked demands by all classes
BD_c	Number of blocked demands by class 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 classes
AD_c	Number of accepted demands by class 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
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
SH_i	Squatted resources from higher priority $class_i$
SL_i	Squatted resources from lower priority $class_i$
K_i	Kicked resources from lower priority $class_i$

4. Detailed Review on Alternative Resource Allocation Models

This section is divided into two subsections, Resource Constraints Models (RCMs) such as BAMs, and Non Constrained

Models (NCMs) such as First-In-First-Out (FIFO) as follows:

4.1. 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 [10]. 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 [34]. For the sake of keeping compatibility with RFCs, from 4125 to 4128 [19, 20, 46, 47], 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 [19], the shares for each class are not isolated. Consequently, the existence of the cross-allocated bandwidth resource cannot be obviated.

4.1.1. Maximum Allocation Model (MAM)

MAM is described in [19]. 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. 1.

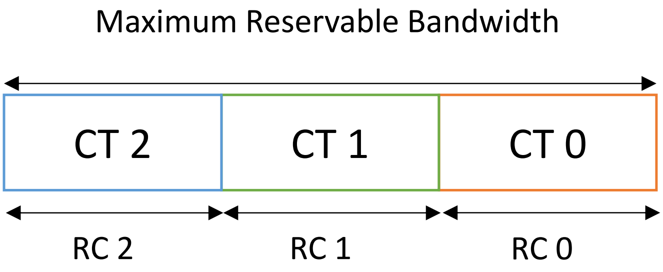


Figure 1: MAM allocation model

MAM 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. (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. (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). Moreover, the sum of resource allocations of TC always corresponds to the resources available for allocation on link considered with a constraint:

$$S_c <= RC_c <= R \tag{1}$$

$$\sum_{c=0}^{N-1} S_c <= R \tag{2}$$

Finally

$$\sum_{c=0}^{N-1} RC_c >= R \tag{3}$$

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.

4.1.2. Russian Doll Model (RDM)

RDM is described in [20]. It presents a more sophisticated technique for bandwidth resource sharing among classes than MAM as shown in Fig. 2.

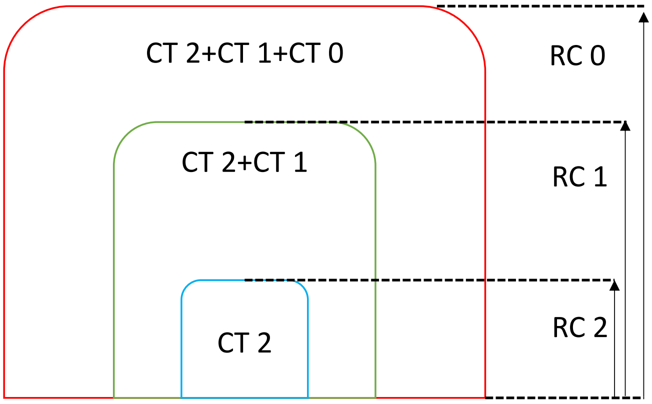


Figure 2: RDM allocation model.

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 ≤ c ≤ 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.
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. 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. (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 $i \in [0, N-1]$

$$\sum_{j=i}^{c-1} S_j <= RC_i <= R \tag{4}$$

   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. (5).

$$\sum_{i=0}^{N-1} S_i <= R \tag{5}$$

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. (6).

$$\begin{aligned}(UR)_i = min[RC_c - \sum S(CT_b, h)\ for\ h \leq P\ and\ c \leq b \leq 7,\\ RC_{(c-1)} - \sum S(CT_b, h)\ for\ h \leq P\ and\ c \leq b \leq 7,\\ \dots,\\ RC_0 - \sum S(CT_b, h)\ for\ h \leq P\ and\ c \leq b \leq 7]\end{aligned} \tag{6}$$

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 [20] 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.

4.1.3. AllocTC-sharing Model (AllocTC)

AllocTC is described in [22]. 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.

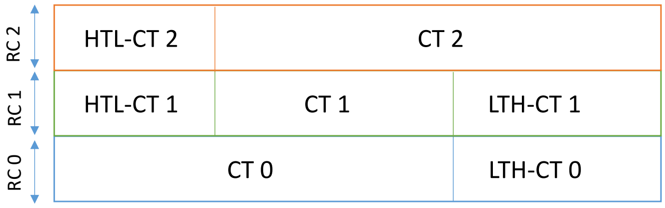


Figure 3: AllocTC-sharing allocation model.

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

- Loan [1] "HTL" in this configurable allocation method, is

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

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. (7) and Eq. (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.

$$HTL_i <= RC_i \ e \ LTH_i <= RC_i \tag{7}$$

$$MAX(S_i) <= RC_i + \sum_{j=i+1}^{N-1} LTH_j + \sum_{k=0}^{i-1} HTL_k \tag{8}$$

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.

4.2. Non Constrained Models

FIFO model is described in [48]. 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. 4.

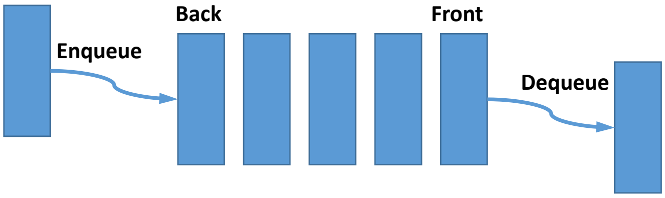


Figure 4: FIFO model.

FIFO is an approach for handling the demands so that the 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.

5. 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.

5.1. 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. 5.

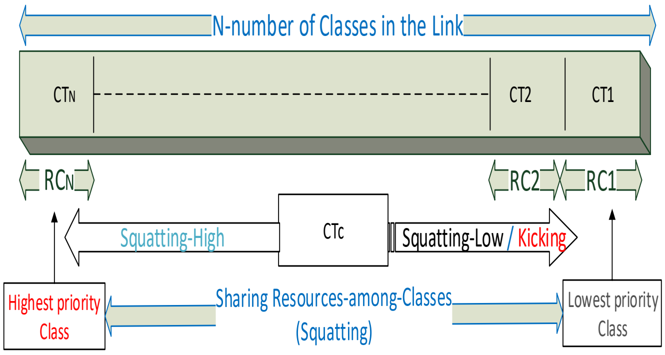


Figure 5: SKM-Strategy

5.2. 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.

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.

$$S_c \leq RC_c \tag{9}$$

$$\sum_{c=1}^{N} RC_c = R \tag{10}$$

Algorithm 1 Process Assignment algorithm for SKM

```
procedure PROCESS ASSIGNMENT(Loop $D$ :Demands; Loop Demands)
    for Each Demand $d_l = d_l(CT_i) \in D$ do
        if $d_l \le RC_i$ then                                        ▷ Strategy MAM
            Allocate $d_l$ resources from the class i
        else if $\exists j \quad s.t. \quad j > i \wedge d_l \le RC_j - min(S_j, RC_j)$ then    ▷ Strategy RDM or Squatting-High
            Allocate $d_l$ resources from $CT_j$                      ▷ $SH_j$
        else if $\exists j$ where $j < i \quad s.t. \quad d_l \le RC_j - min(S_j, RC_j)$ then    ▷ Squatting-Low
            Allocate $d_l$ resources from $(CT_j)$                    ▷ $SL_j$
        else
            found-kick=false
            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 $(CT_j)$
                    found-kick=true
                end if
            end for
            if ¬(found-kick) then
                Reject $d_l$
            end if
        end if
    end for
end procedure
```

Eq. (9) ensures that the resources needed to serve the already existing demands plus the new demand do not exceed class resources constraint while Eq. (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. (11), Eq. (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.

$$S_c \le RC_c + \sum\nolimits_{i=c+1}^{N} SH_i \tag{11}$$

$$SH_i \le RC_i - min(S_i, RC_i) \tag{12}$$

Step 3 (Squatting-Low): Try to squat unused resources starting from the lower adjacent priority 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. (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. (14). Note that the lowest priority class cannot use Squatting-Low strategy.

$$S_c \le RC_c + \sum\nolimits_{i=c+1}^{N} SH_i + \sum\nolimits_{i=1}^{c-1} SL_i \tag{13}$$

$$SL_i \le RC_i - min(S_i, RC_i) \tag{14}$$

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. (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. (16). Note that the lowest priority class cannot use kicking strategy.

$$S_c \le RC_c + \sum\nolimits_{i=c+1}^{N} SH_i + \sum\nolimits_{i=1}^{c-1} SL_i + \sum\nolimits_{i=1}^{c-1} K_i \tag{15}$$

$$K_i \le RC_i \tag{16}$$

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.

6. 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.

6.1. Technical behavior and other operational characteristics

Table 3 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.

Table 3: 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

6.1.1. *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. (17-22) as below:

$$AR = AD/D \tag{17}$$

$$AR_c = AD_c/D_c \tag{18}$$

$$Bp = BD/D \tag{19}$$

$$Bp_c = BD_c/D_c \tag{20}$$

$$U = \frac{\sum_{j=1}^{D} d_j(CT_c)\, I_{A(j)}}{R} \tag{21}$$

$$U_c = \frac{\sum_{j=1}^{D_c} d_j(CT_c)\, I_{A_c(j)}}{R} \tag{22}$$

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. Please recall that, AR, AR_c, Bp, Bp_c, U, U_c, AD, AD_c, BD, BD_c, D, D_c, $d_j(CT_c)$, R and RC_c definitions are given in Table 1 and Table 2.

6.2. *Offline SKM Behavior*

Fig. 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.

6.2.1. *Example of Proposed Offline SKM Algorithm*

Fig. 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:

#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 4 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,

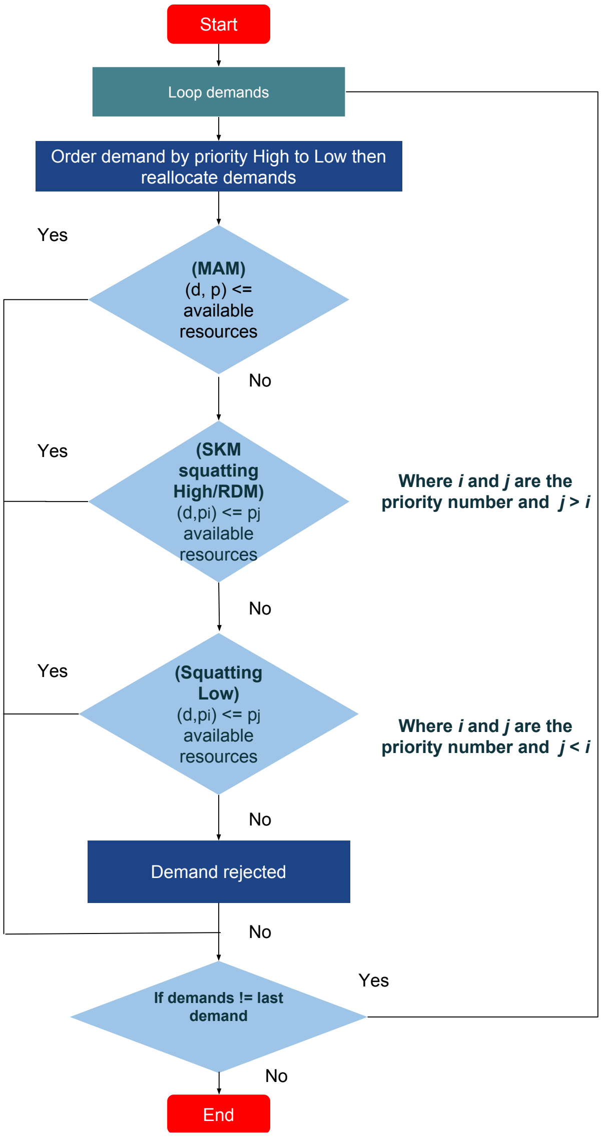

Start
Loop demands
Order demand by priority High to Low then reallocate demands
Yes
(MAM) (d, p) <= available resources
No
Yes
(SKM squatting High/RDM) (d,pi) <= pj available resources
Where i and j are the priority number and j > i
No
Yes
(Squatting Low) (d,pi) <= pj available resources
Where i and j are the priority number and j < i
No
Demand rejected
No
Yes
If demands != last demand
No
End

Figure 6: SKM Offline

which used ten units from its priority class resources and borrowed two unused units from class 3 resources. Table 5 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.

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. (23).

$$AR(T) = \sum_{\forall t \in T} \frac{(AD)_t}{D_t + (AD)_{t-1}} \times 100 \tag{23}$$

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.(24).

$$Bp(T) = \sum_{\forall t \in T} \frac{(BD)_t}{D_t + (BD)_{t-1}} \times 100 \tag{24}$$

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.(25).

$$AR_c(T) = \sum_{\forall t \in T} \frac{(AD_c)_t}{D_{c_t} + (AD_c)_{t-1}} \times 100 \tag{25}$$

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.(26).

$$Bp_c(T) = \sum_{\forall t \in T} \frac{(BD_c)_t}{D_{c_t} + (BD_c)_{t-1}} \times 100 \tag{26}$$

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.(27).

$$U(T) = \frac{\sum_{j=1}^{D} d_j(CT_c)\, I_{A(j)}\, T_j}{R * T} \times 100 \tag{27}$$

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

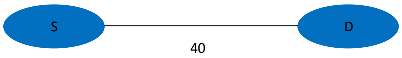


Figure 7: Single-link

Table 4: 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 5: 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

capacity of resources of the same class at the time of observation Eq.(28).

$$U_c(T) = \frac{\sum_{j=1}^{D_c} d_j(CT_c)\, I_{A_c(j)}\, T_j}{R * T} \times 100 \tag{28}$$

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.

6.3. *Online SKM Behavior*

Fig. 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.

Algorithm 2 Resource assignment algorithm for SKM Online

```
procedure SKMOnline($D$ :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 for
end procedure
```

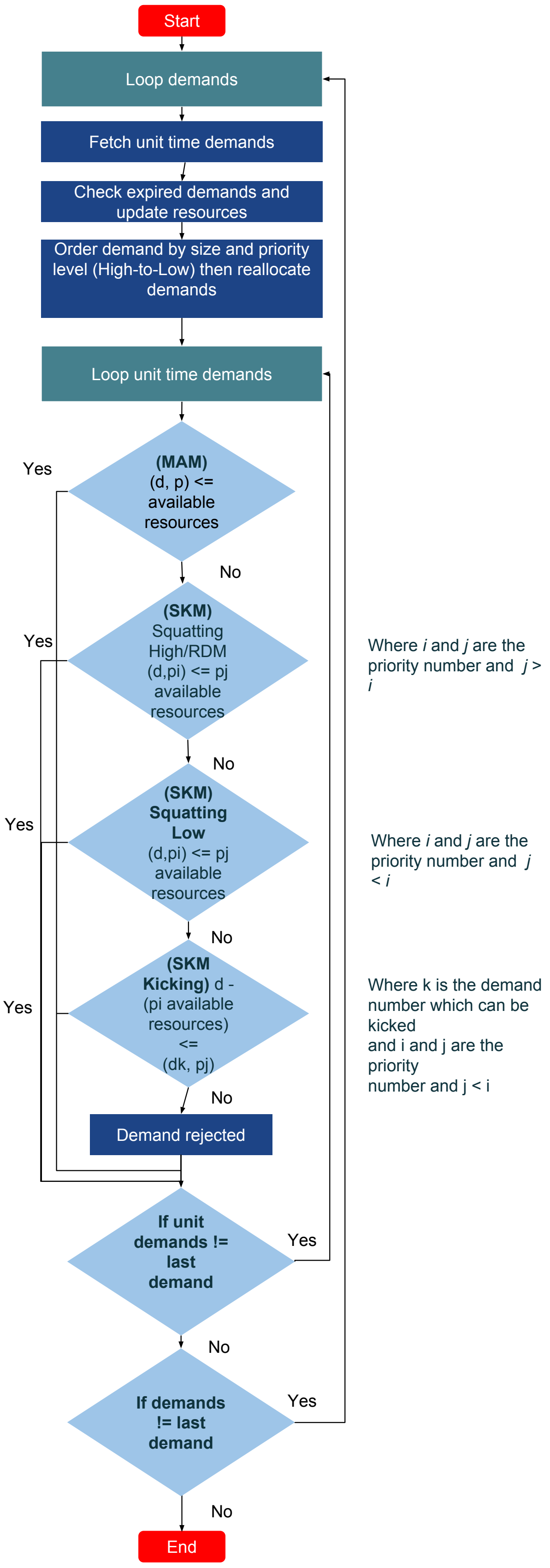

Start
Loop demands
Fetch unit time demands
Check expired demands and update resources
Order demand by size and priority level (High-to-Low) then reallocate demands
Loop unit time demands
Yes
(MAM) (d, p) <= available resources
No
Yes
(SKM) Squatting High/RDM (d,pi) <= pj available resources
Where i and j are the priority number and j > i
No
Yes
(SKM) Squatting Low (d,pi) <= pj available resources
Where i and j are the priority number and j < i
No
Yes
(SKM Kicking) d - (pi available resources) <= (dk, pj)
Where k is the demand number which can be kicked and i and j are the priority number and j < i
No
Demand rejected
If unit demands != last demand
Yes
No
If demands != last demand
Yes
No
End

Figure 8: SKM Online

6.3.1. 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 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
#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 6 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 7 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.

6.4. 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 paper:

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 [22], as explained in 6.5.
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:

- Scenario two: traffic load generated is the same for TCs of all priorities as detailed in 6.6.1.
- Scenario three: traffic load generated is higher for TCs of higher priority as detailed in 6.6.1.
- Scenario four: traffic load generated is high for TCs of lower priority as detailed in 6.6.1.
- scenario five: we also tested the impact of varying number demand lifetime on SKM performance as detailed in 6.9.1.

6.5. 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.

6.5.1. 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 [22] 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 8 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.

Table 6: SKM example (Online)

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

Table 7: 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

- 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)
- Simulation stop criteria: number of demands

Table 8: 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

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.

6.5.2. *Scenario one 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. 10.

Fig. 10a 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. 10b and Fig. 10c) 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. 10d 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.

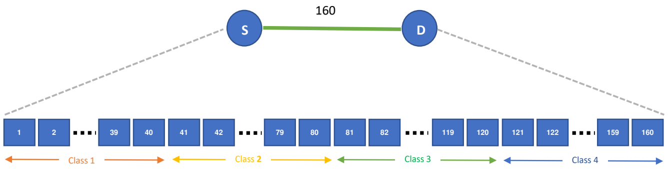


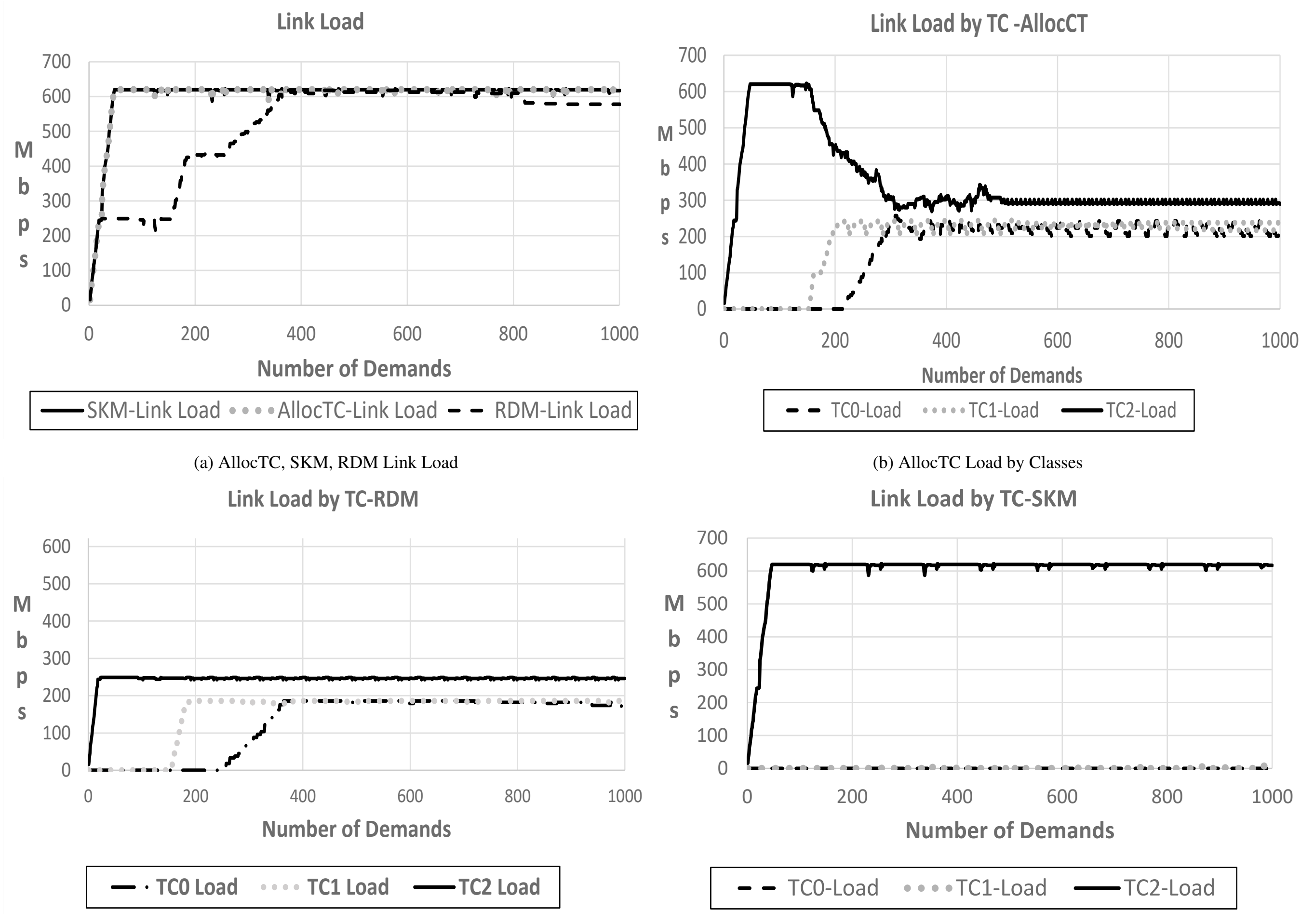
Figure 9: Proof-of-Concept-Simulated Topology

6.6. *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. 9 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.

6.6.1. *Simulation scenario settings two, three, four*

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 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 9 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 follows:



(a) AllocTC, SKM, RDM Link Load

(b) AllocTC Load by Classes

(c) RDM Load by Classes

(d) SKM Load by Classes

Figure 10: Comparison of Link Load and Link Load per Class in scenario one

Table 9: 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

- 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.

6.7. Obtained 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. 11-14.

6.7.1. Scenario two Description and Results Evaluation

In this simulation scenario, Table 10 shows the summary of the obtained results by each model from Figs. 11a - 11f in terms of the metrics U, AR, U_c and AR_c. Table 10 also shows the numerical estimations (expected metric values).
The expected value of U_c for each algorithm is evaluated and presented in Eq.(29) as follows:

$$\mathbf{E}[U_c] = \frac{D \times p_c}{R} \tag{29}$$

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.(30) as follows:

$$\mathbf{E}[U] = \frac{\sum_{c=1}^{N} \mathbf{E}[U_c]}{R} \tag{30}$$

The expected value of AR_c for each algorithm is evaluated and presented in Eq.(31) as follows:

$$\mathbf{E}[AR_c] = \frac{D \times p_c}{D_c} \tag{31}$$

The expected value of AR for each algorithm is evaluated and presented in Eq.(32) as follows

$$\mathbf{E}[AR] = \frac{\sum_{c=1}^{N} \mathbf{E}[U_c]}{D} \tag{32}$$

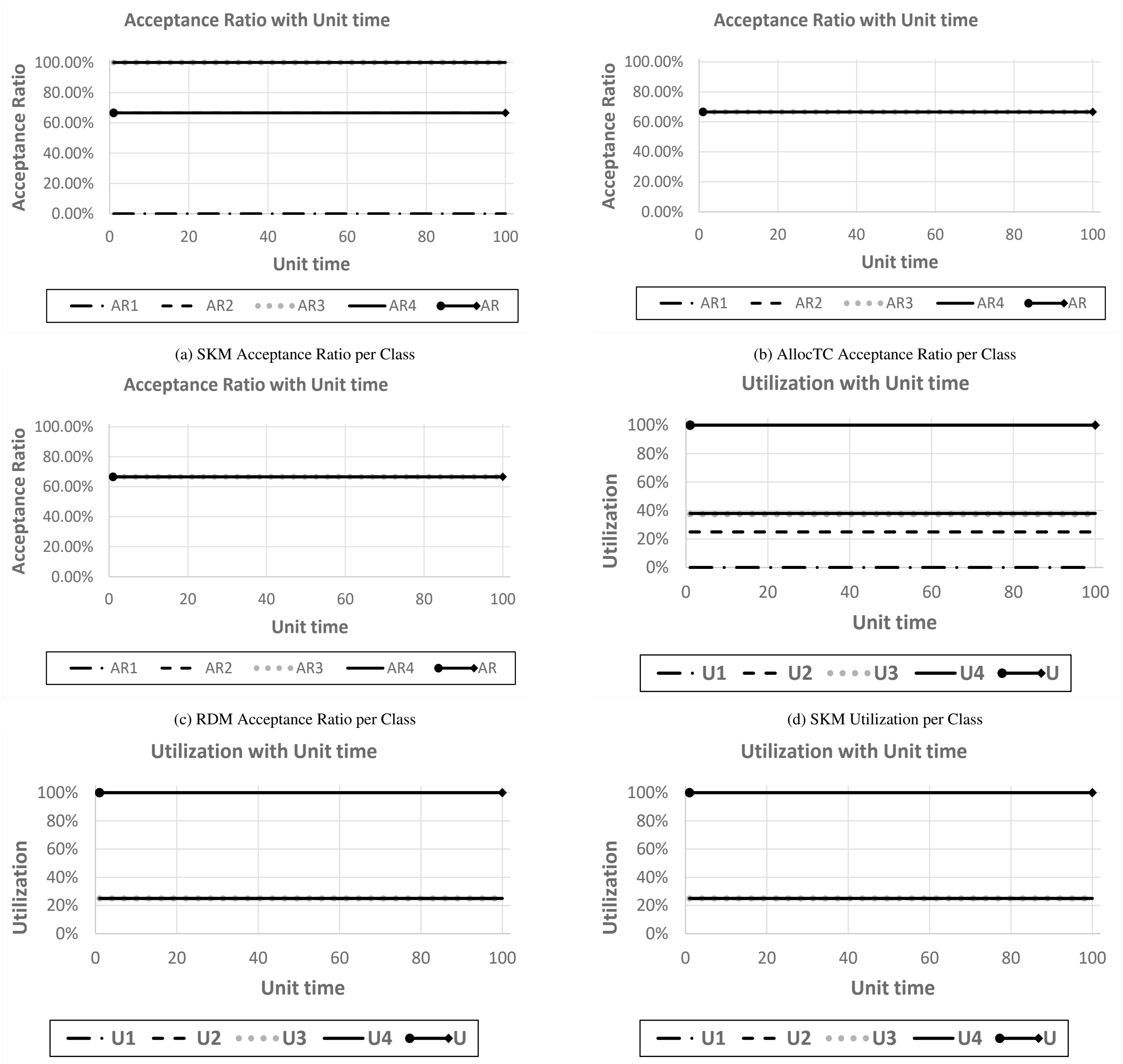
This proves that our simulations performance gives similar results to the numerical estimations. Please note that utilization per class in terms of numerical estimations is calculated in general, from the expected performance of each class according to applied strategy. On the contrary, the acceptance ratio is calculated on specific cases depending on the number of demands in each class in each scenario.

As shown in Fig. 11 and Table. 10, 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. 11a - 11c and Table. 10 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. 11d - 11f and Table. 10).

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 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.

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 9). 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.

SKM achieves the lowest performance in lower classes due to the kicking operation which results in the expelling of the



(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 11: Comparison of Utilization and Acceptance Ratio per Class in scenario two

Table 10: 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

Table 11: 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%

lower priority users in order to satisfy the demand requirements of the high priority classes as shown in Table. 10, Fig. 11a and Fig. 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.

6.7.2. Scenario three Description and Results Evaluation

In this simulation scenario, Table. 11 shows the summary of the obtained results by each model in terms of utilization and acceptance ratio from Figs. 12a - 12f and compares it with the expected results.

Fig. 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 9. 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. 12a- 12f and Table. 11. 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. 11). 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 9.

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.

6.8. Scenario four Description and Results Evaluation

In this simulation scenario, Figs. 13a and 13b reflect the behavior of each algorithm in terms of U_c, U, AR_C and AR along 100 unit times.

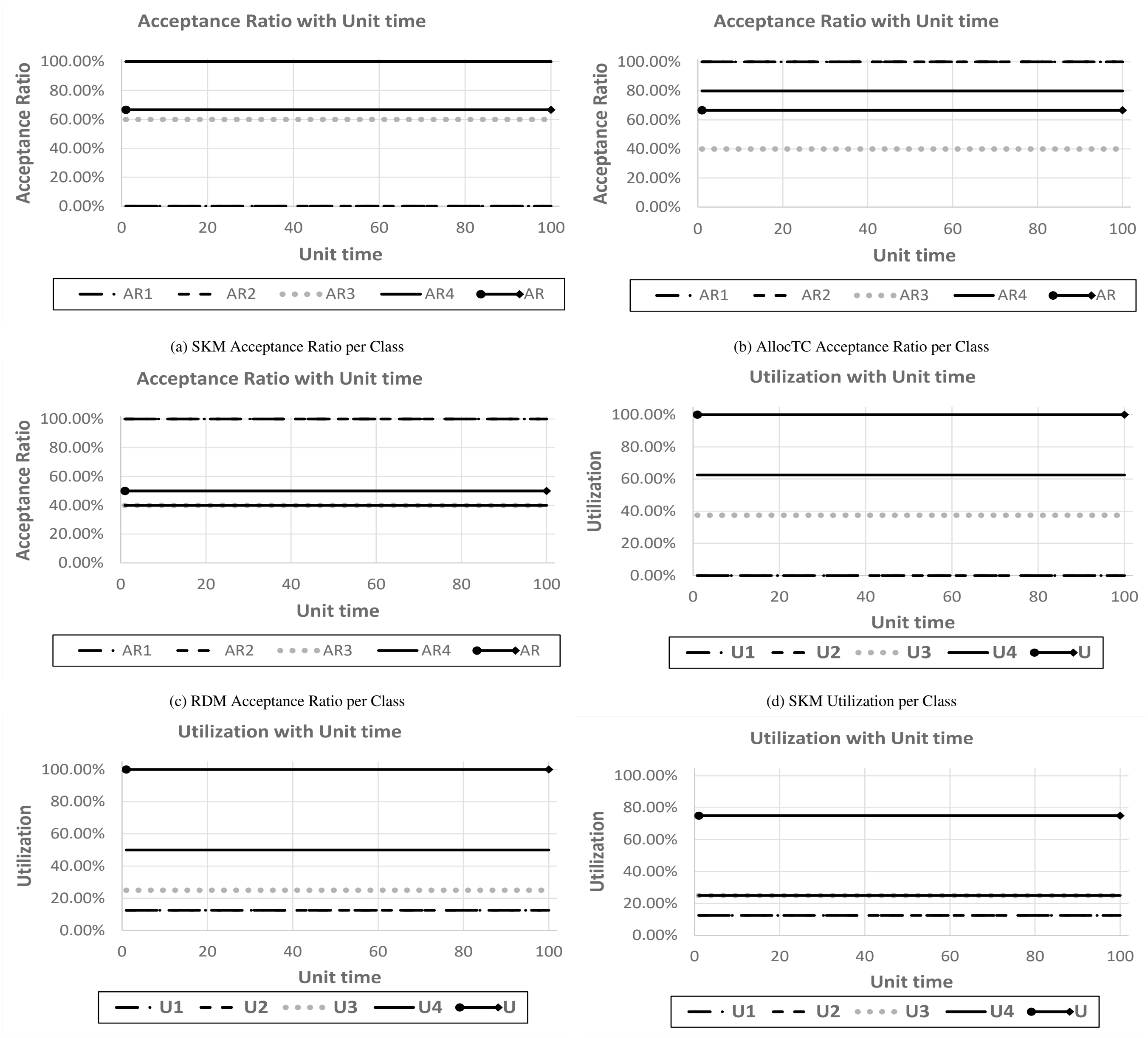
Simulations for scenario four showed that SKM, AllocTC and RDM have similar behaviour for traffic patterns in which lower priority classes have greater demands for resources. This is as expected from the performance of each algorithm since the lower classes can share all unused resources from the higher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

6.9. 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.

6.9.1. Simulation scenario five setting

In the simulations, the demands are generated with a lifetime of each demand varied randomly from 1 to 100-time slots and



(a) SKM Acceptance Ratio per Class

(b) AllocTC Acceptance Ratio per Class

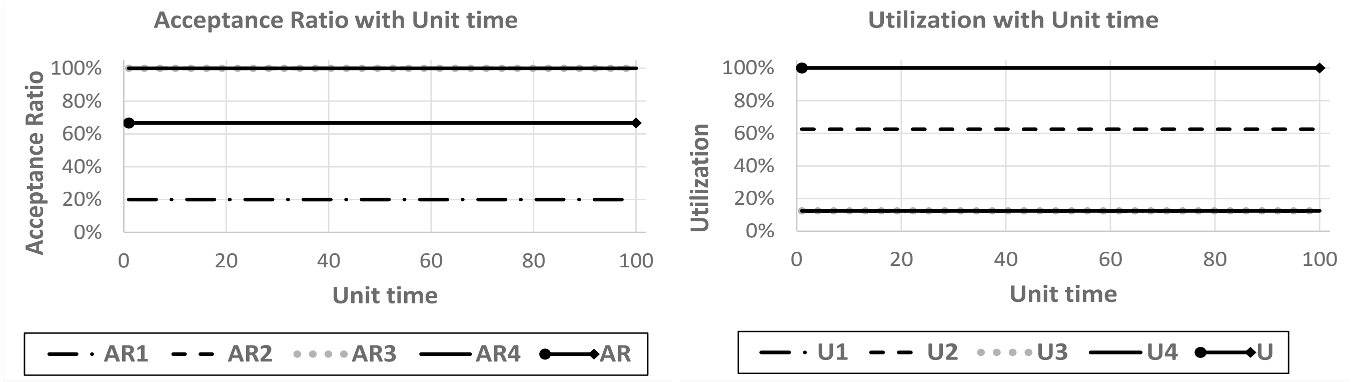
(c) RDM Acceptance Ratio per Class

(d) SKM Utilization per Class

(e) AllocTC Utilization per Class

(f) RDM Utilization per Class

Figure 12: Comparison of Utilization and Acceptance Ratio per Class in scenario three



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

(b) SKM, AllocTC, RDM Utilization per Class

Figure 13: Comparison of Utilization and Acceptance Ratio per Class in scenario four

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. 12 shows the summary of the simulation scenario 5.

Table 12: 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.

6.9.2. *Scenario five Description and Results Evaluation*

In this simulation scenario, Table 13 shows the summary of the obtained results by each model in terms of utilization and acceptance ratio from Figs. 14a - 14f in terms of the results from simulations for the the metrics U, U_c, AR, AR_c. From Fig. 14 and the shown Table 13, 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. 14b, Fig. 14d, Fig. 14f and Fig. 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 13. 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. 14b, Fig. 14c, Fig. 14f and Fig. 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.

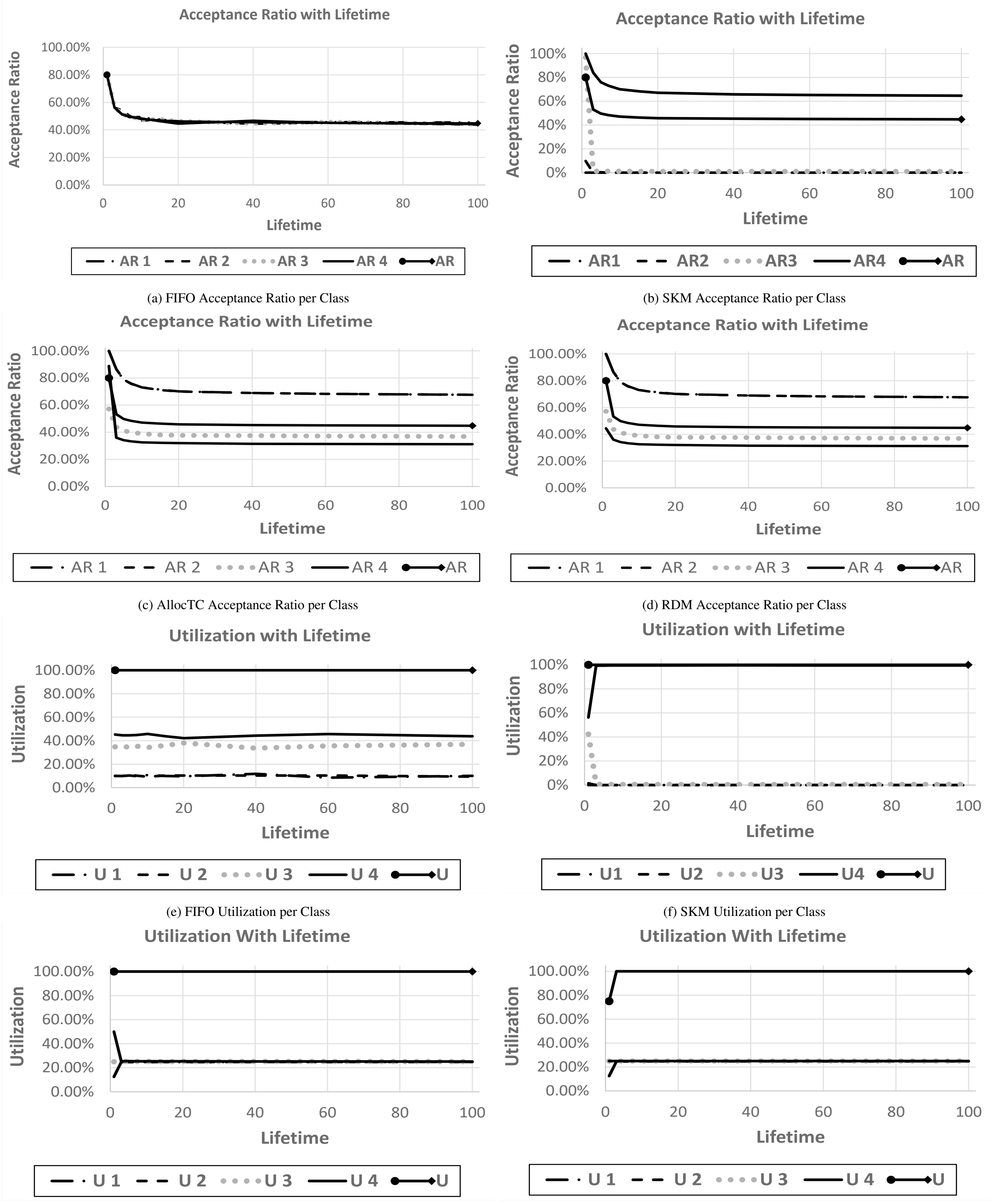
As expected, in this scenario SKM outperforms FIFO in terms of U_4 and AR_4 by 50.56% and 21.72% respectively (see Fig. 14b, Fig. 14a, Fig. 14f and Fig. 14e). FIFO does not consider classes so it cannot provide QoS. Moreover, FIFO will not result in the same utilization and acceptance ratio across the 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.

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.

7. 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 14 shows the summary of the obtained results from each strategy from Fig. 14 in terms of AR tendency (i.e., where AR values converge).

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. 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



(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 14: Comparison of Utilization and Acceptance Ratio per Class in scenario five

Table 13: 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%

Table 14: 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%

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 14 and Fig. 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. 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 14 and Fig. 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. 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 14 and Fig. 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. 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 14 and Fig. 14a).

8. 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. 11 and Fig. 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. 11). 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. 10).

- 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. 10 and Table. 11).

- 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. 10, Table. 11 and Table. 13).

- 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. 13).

- 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. 11 and Fig. 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. 11 and Fig. 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. 11 and Fig. 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. 11 and Fig. 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.

9. CONCLUSIONS AND FUTURE WORK

BAMs are of great value in the context of efficient and customized use of network resources. Therefore, in this paper, 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 paper 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.

As a future extension, we intend to include full network topology to study the SKM performance in the paths to optimally allocate demand of available resources across the network. As another future work, SKM will be improved by considering aforementioned thresholds to define and guarantee minimum resources for each class that will avoid resources beat down for lower priority classes. Moreover, we want to mention that other parameters such as delay can be integrated into our proposed algorithm. Last but not least, SKM can also be adapted to the allocation of node resources with minimal modification, which we consider as future work.

10. Acknowledgment

This work has been partially supported by the Ministerio de Economy Competitividad of the Spanish Government under project TEC2016-76795-C6-1-R and AEI/FEDER, UE, and the SGR project, grant number 2017 SGR 397, from the Generalitat de Catalunya.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] C.Marquez et al., “Resource Sharing Efficiency in Network Slicing,” IEEE Transactions on Network and Service Management (TNSM), 2019. DOI:10.1109/TNSM.2019.2923265.

[7] C.Marquez et al., “How Should I Slice My Network: A Multi-Service Empirical Evaluation of Resource Sharing Efficiency,” ACM MOBICOM, 2018. DOI: 10.1145/3241539.3241567.

[8] 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. https://ieeexplore-ieee-org/document/8427129.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

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

[33] 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.

[34] 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.

[35] 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.

[36] S.K. Sadon, N.M. Din, M.H. Al-Mansoori, N.A. Radzi, I.S. Mustafa, M. Yaacob and M.S.A. Majid, “Dynamic hierarchical bandwidth allocation using Russian Doll Model in EPON,” Comput. Electr. Eng. 38 (6) (2012) 1480-1489. DOI:10.1016/j.compeleceng.2012.05.002.

[37] 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.

[38] J. Socrates-Dantas, R. Melo Silveira, D. Careglio, J. Roberto Amazonas, J. SolePareta 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.

[39] 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.

[40] 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.

[41] 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, 2018, pp. 1-10. DOI:10.1109/ICTON.2018.8473869.

[42] A. El-mekkawi, X. Hesselbach and J. R. Piney, "A Novel Admission Control Scheme for Network Slicing based on Squatting and Kicking Strategies," 2019 12th International Conference on Transparent Optical Networks (JITEL), Zaragoza, 2019, pp. 1-8. https://easychair.org/publications/preprint/6Ccb.

[43] 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.

[44] 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.

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

[46] 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.

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

[48] 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.