Flexible Resource Allocation for eMBB and mMTC services in a Time-Varying Satellite Topology

Muhammad Ahsan, Thang X. Vu, and Symeon Chatzinatos
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg
Email:{muhammad.ahsan, thang.vu, symeon.chatzinotas} @uni.lu

***Abstract*—The incorporated terrestrial and non-terrestrial networks (NTNs) in 6G networks are essential to provide global connectivity and support heterogeneous quality of service (QoS) requirements. Network slicing (NS) is the key technological enabler to realize these requirements by provisioning various services over the same physical infrastructure. While showing outstanding performance in terrestrial networks, having efficient NS deployment in NTNs is challenging due to the highly dynamic nature of NTN topology. In this paper, we investigate network slicing in dynamic environment of the integrated low earth orbit (LEO)-terrestrial network serving distinct services, namely enhanced mobile broadband (eMBB) and massive machine-type communications (mMTC). In particular, we propose a flexible resource allocation routing framework and introduce a multi-scale time slot concept to efficiently embed mMTC and eMBB services considering the very fast variation of LEO topology. A mathematical model based on mixed integer linear programming (MILP) is proposed with the objective to maximize the number of served requests subjected to services' deadlines and topology dynamics. To tackle the high complexity of the formulated optimization problem, an iterative algorithm based on successive convex approximation (SCA) is proposed. The simulation results reveal that the proposed solution outperforms the benchmark schemes in terms of percentage of served requests and average eMBB sum rate.**

***Index Terms*—6G, eMBB, Flexible Resource Allocation, LEO satellites, mMTC, Network Slicing, Satellite Network.**

I. Introduction

The sixth generation (6G) communication network aims to provide ubiquitous connectivity for billions of devices across the globe [1]. The integration of non-terrestrial networks (NTNs) with the existing terrestrial networks (TNs) will play a crucial role to meet the goals envisioned by 6G networks [2]. The low earth orbit (LEO) satellites belong to the class of non-geostationary satellite orbit (NGSO) and operate at an altitude of 350 km to 2000 km [3]. LEO satellites operate in multiple orbits and are connected to the ground gateways via ground-to-satellite links (GSLs), while each satellite establishes multiple connections with the satellites in the same and neighboring orbits known as the inter-satellite links (ISLs) as shown in Fig. 1. LEO satellites offer affordable round-trip latency, and are well-suited to cater 6G services with diverse requirements [4]. These services include enhanced mobile broadband (eMBB) and massive machine type communication (mMTC). The eMBB services require a constant data stream and are more demanding in terms of bandwidth [5]. The mMTC services include a large number of connected devices that require resources for a short amount of time due to their strict latency requirement [6]. Using Network Function Virtualization (NFV), network slicing can accommodate these distinct service types over a common physical infrastructure [7]. In addition, network slicing enables flexible resource provisioning as each slice is allocated a chunk of available resources according to its quality of service (QoS) requirement [8]. The application of network slicing on LEO satellites faces a dynamic topology challenge due to the swift movement of LEO satellites [9]. Further, the available resources in LEO satellite network are scarce which makes it more challenging to accommodate applications requiring high bandwidth [10].

In recent years, routing and resource allocation in integrated TN-NTN networks has attracted much research attention. For example, the authors in [11] proposed a routing and resource allocation strategy for IoT slices over CubeSATs network. They presented a heuristic algorithm for slice admittance and MILP based mathematical model for resource allocation with the objective to minimize the service level agreement (SLA) violation. The proposed work therein, however, only considered static virtual nodes (VNs) for topology construction. The authors in [12] investigated virtual network function (VNF) placement in an integrated space-air-ground network consisting of high elliptical orbit (HEO) in space segment and LEO-medium earth orbit (MEO) in the air segment. A heuristic algorithm was proposed therein considering a static satellite topology with the aim to map the VNFs on path with minimum delay. Mario *et al.* in [13] presented a dynamic routing technique over MEO satellites where they used the topology snapshots in different time slots to map the virtual network request (VNR). They formulated mathematical model based on MILP with the objective to minimize the path migration cost and solved it with relaxation algorithm. However, [13] applies to only single type of service. Literature [14] proposed a dynamic intersatellite path computation strategy where the path between two satellites was selected by constructing time-space graph based on the link attributes at different time intervals. The work in [15] introduced a dynamic virtual network embedding (VNE) approach for nodes and links mapping in an integrated MEO-Terrestrial network topology. The mapping was done based on request duration spanning multiple time slots, but this work considered the arrival of requests at start of the time slots and each request occupied the resources for the entire duration of a time slot.

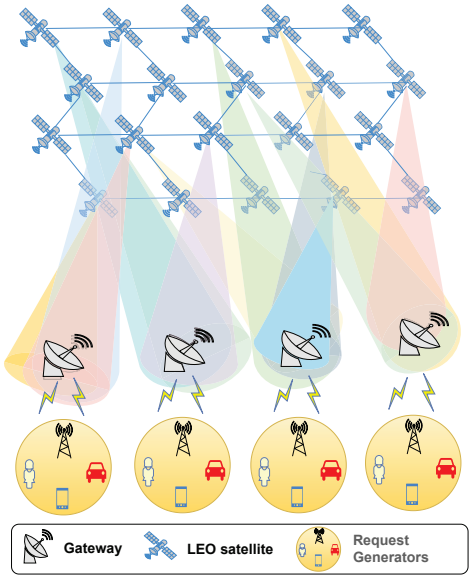

Fig. 1: The integrated TN-NTN architecture.

Most of the above studies focus on static or dynamic routing and execute the resource allocation for different slices by exploiting the priority-based solutions to tackle different QoS requirements. While these methods can provide reasonably good performance if the slices' requirements are relatively close, as will be shown later, their performance can be significantly degraded if the slices' QoS requirements are highly heterogeneous, e.g., eMBB and mMTC services. To tackle this issue, we propose a novel flexible resource allocation scheme for eMBB and mMTC services in a time-varying LEO satellite topology. Our contributions are summarized as follows:

- To efficiently address the highly distinct QoS requirements of eMBB and mMTC services, we introduce a sub-slot concept for modelling the arrival of mMTC requests. Since the mMTC requests' arrival times are puncture and their life times are much shorter than eMBB requests, the proposed sub-slot concept allows a more efficient deployment of the time-bandwidth resources.
- We formulate the joint resource allocation and routing for eMBB and mMTC services as a mixed-integer linear programming (MILP) optimization problem with the objective of maximizing the number of served requests subjected to practical network constraints involving the topology dynamics, the mMTC rates computed based on the required latency, and eMBB capacity.
- To tackle the exponential computation complexity of the formulated optimization problem, we propose a successive convex approximation (SCA)-based iterative algorithm to solve the relax optimization problem. A proper penalty function is added to the objective function of the relax problem to automatically tighten the performance gap to the original objective function.
- Finally, we compare the performance of proposed technique with benchmark schemes from recent literature via simulation results and show our superior performance in terms of the percentage of served requests and average eMBB sum rate.

II. System Model

A. Network Topology

We consider a LEO satellite network providing heterogeneous eMBB and mMTC services in a given geographical area. For ease of presentation, the overall satellite service duration is divided into multiple configuration periods, denoted as time slots and are indexed by t. The resource allocation and routing rules are updated at beginning of every time slot. The network topology is assumed static within one time slot, but dynamically changes from one time slot to another. The considered network consists of space and ground segments and is represented by $G^t(\mathcal{N}, \mathcal{E}(t))$, where $\mathcal{N}$ is a set of nodes consisting of LEO satellites in the space segment and gateways in the ground segment. $\mathcal{E}(t)$ is a set of active edges during time slot t, where the capacity of each edge is indicated with $c_e(t)$. By nature, the LEO satellite topology is time-varying, resulting in changes in ISLs and GSLs in different time slots. We employ the widely-used Walker-Delta constellation to model the topology dynamics [16]. In this model, each satellite maintains 4 ISLs, including two in-orbit connections and two inter-orbit connections. Furthermore, the GSLs may change depending on the position of satellites with respect to the gateways. For each time slot t, let $\mathcal{P}(t)$ denote a set of accessible paths between all the gateways through the satellite network, the capacity of each path is represented by $c_p(t)$. Since the LEO satellite orbits are time-varying but predictable, $\mathcal{P}(t)$ can be computed before hand and is available at beginning of time slot t.

B. Traffic modelling

We adopt the widely used Poisson point process (PPP) to model the requests arriving at each time slot. Let $\mathcal{R}^E(t)$ and $\mathcal{R}^M(t)$ denote the set of arriving eMBB and mMTC requests at time slot t, respectively. By definition, we obtain $|\mathcal{R}^E(t)| \sim \text{ppp}(\lambda_e)$ and $|\mathcal{R}^M(t)| \sim \text{ppp}(\lambda_m)$, where λ_e and λ_m are the mean arrival rate of eMBB and mMTC requests, respectively. Due to the nature of eMBB services, we assume that all eMBB requests are required to be served at the beginning of each time slot. Thus, each eMBB request $i \in \mathcal{R}^E(t)$ can be described by its destination node D_i, minimum rate requirement c_i^E, data size s_i^E and maximum life time in time slots η_i, i.e., $i = \{D_i, c_i^E, s_i^E, \eta_i\}$. On the other hand, mMTC requests are more punctual with a much shorter life time and not all mMTC requests start at the beginning of each time slot. Therefore, we introduce the concept of sub-slot to better capture the mMTC services. Each time slot duration is divided into L sub-slots. We model each mMTC request $j \in \mathcal{R}^M(t)$ by the destination D_j, the start time l_j, the data size s_j^M and the lifetime ζ_j in sub-slots, i.e., $j = \{D_j, l_j, s_j^M, \zeta_j\}$.

C. Variables and Achievable eMBB Throughput

For ease of presentation, a summary of variables and parameters used in this paper is provided in Table I. For

eMBB resource allocation, let $x_{i,p}(t) \in \{0,1\}$ denote the binary variable mapping the eMBB request i to the accessible path p at time slot t. Since the mMTC services' lifetime is usually shorter than the time slot duration, assigning the mMTC services to a path for the whole time slot duration is not efficient. Therefore, we divide each time slot duration into multiple sub-slots and denote $y_{j,p,l}(t) \in \{0,1\}$ as the binary variable mapping the mMTC request j to the path p for sub-slot l of time slot t. The benefit of this sub-slot concept allows to harvest the residual link capacity after mMTC requests are served.

TABLE I: Summary of main notations and variables

Variables	Meaning
$x_{i,p}(t)$	$x_{i,p}(t) = 1$, when an eMBB request i is routed through path p during time slot t.
$y_{j,p,l}(t)$	$y_{j,p,l}(t) = 1$ when a mMTC request j is routed through path p in the sub-slot $l \in t$.
Parameters	**Meaning**
$\mathcal{R}^E(t)$	set of eMBB requests arriving at t.
$\mathcal{R}_l^M(t)$	set of mMTC arriving at sub-slot l within t.
$\mathcal{P}(t)$	set of computed paths during t.
$\mathcal{E}(t)$	set of edges at time slot t.
$\mathcal{P}_e(t)$	set of paths passing through edge e at t.
η_i	no. of time slots required for eMBB request i.
ζ_j	no. of sub-slots required for mMTC request j.
s_i^E, s_j^M	size of eMBB and mMTC request i and j.
c_j^M	required capacity for mMTC request j.
c_i^E	required capacity for eMBB request i.
$c_e(t), c_p(t)$	capacity of edge e, path p during time slot t.
Δ_t, Δ_l	duration of time slot and sub-slot.

Given the size and lifetime, the required rate for the mMTC request j is given as $c_j^M = \frac{s_j^M}{\zeta_j}$. Thus, the capacity of path p at sub-slot l allocated to the mMTC requests can be computed as:

$$r_{p,l}^M(t) = \sum\nolimits_{j \in \mathcal{R}_l^M(t)} c_j^M y_{j,p,l}(t), \forall l, p, \tag{1}$$

where $\mathcal{R}_l^M(t)$ is the set of mMTC requests during sub-slot l.

Denote $r_{p,l}^E(t)$ as the available capacity on path p during a sub-slot after serving mMTC requests. In principle, $r_{p,l}^E(t)$ will be used to embed eMBB requests, which is calculated as follows:

$$r_{p,l}^E(t) = c_p(t) - r_{p,l}^M(t) = c_p(t) - \sum_{j \in \mathcal{R}_l^M(t)} c_j^M y_{j,p,l}(t), \tag{2}$$

where $c_p(t)$ is the capacity of path p.

The achievable throughput (in bits per time slot) for eMBB service during time slot t on path p is hence computed as:

$$\begin{aligned} R_p^E(t) &= \sum_l \Delta_l r_{p,l}^E(t) \\ &= \Delta_t c_p(t) - \sum\nolimits_{j \in \mathcal{R}_l^M(t)} \sum\nolimits_{l=1}^{L} \Delta_l c_j^M y_{j,p,l}(t), \end{aligned} \tag{3}$$

where Δ_t and Δ_l are the time slot duration and sub-slot duration, respectively.

D. Problem Formulation

The objective of optimization model is to maximize the number of eMBB and mMTC served requests during each time slot. In addition, the model considers practical constraints for routing of requests related to the eMBB capacity, throughput and mMTC capacity. The optimal path for the routing of each request is obtained as the output of optimization model. A request is routed through a path if the resources are sufficient to satisfy its QoS requirements. The problem is mathematically formulated as follows:

$$\text{OP}: \max_{\mathbf{X},\mathbf{Y}} \left\{ \sum_{p \in \mathcal{P}(t)} \Big(\sum_{i \in \mathcal{R}^E(t)} x_{i,p}(t) + \sum_{l \in t} \sum_{j \in \mathcal{R}_l^M(t)} y_{j,p,l}(t) \Big) \right\} \tag{4}$$

subject to:

$$\text{C1}: x_{i,p}(t) \in \{0,1\}, \ \forall\, i \in \mathcal{R}^E(t), p \in \mathcal{P}(t)$$

$$\text{C2}: y_{j,p,l}(t) \in \{0,1\}, \ \forall\, l, j \in \mathcal{R}_l^M(t), p \in \mathcal{P}(t)$$

$$\text{C3}: \sum\nolimits_{p \in \mathcal{P}(t)} x_{i,p}(t) \leq 1, \ \forall\, i \in \mathcal{R}^E(t)$$

$$\text{C4}: \sum\nolimits_{l=l_j}^{L} \sum\nolimits_{p \in \mathcal{P}(t)} y_{j,p,l}(t) \leq \zeta_j, \ \forall\, j \in \mathcal{R}_l^M(t)$$

$$\text{C5}: \sum\nolimits_{i \in \mathcal{R}^E(t)} \frac{s_i^E}{\eta_i} x_{i,p}(t) \leq R_p^E(t), \ \forall\, p \in \mathcal{P}(t)$$

$$\text{C6}: \sum_{j \in \mathcal{R}_l^M(t)} c_j^M y_{j,p,l}(t) \leq c_p(t), \ \forall\, l \in t, p \in \mathcal{P}(t)$$

$$\text{C7}: \sum\nolimits_{p \in \mathcal{P}_e(t)} \Big(\sum\nolimits_{i \in \mathcal{R}^E(t)} c_i^E x_{i,p}(t) + \sum\nolimits_{l=1}^{L} \sum\nolimits_{j \in \mathcal{R}^M(t)} \frac{\Delta_l c_j^M}{\Delta_t} y_{j,p,l}(t) \Big) \leq c_e(t), \ \forall e,$$

where $\mathbf{X} \triangleq \{x_{i,p}\}_{\forall i,p}$, $\mathbf{Y} \triangleq \{y_{j,p,l}\}_{\forall j,p,l}$ are shorthand notation for embedding variables, and $\mathcal{P}_e(t)$ denotes the set of all paths passing through edge e during time slot t. Here, the constraints C1 and C2 indicate the binary decision variables. Constraint C3 establishes that each eMBB request is routed through one path. Constraint C4 ensures that each mMTC request is allocated required number of sub-slots. Constraint C5 ensures that the path's throughput (in bits per time slot) is sufficient to cater the eMBB requests. The constraint C6 establishes that the capacity of all the mMTC requests passing through a path in a sub-slot is less than the total capacity. Since each edge in the topology can accommodate multiple paths, the embedding of eMBB and mMTC services must also not exceed the capacity of every edge, stated in constraint C7.

III. Proposed Iterative algorithm based on SCA

A. Challenges of solving the problem (4)

The optimization problem defined in (4) is a MILP problem and it is non-convex due to the presence of binary decision variables $\mathbf{X}$ and $\mathbf{Y}$. The problem is NP-Hard in nature and can be solved through branch and bound (BnB) algorithm using commercial solvers such as Gurobi. However, the challenge is the exponential computational complexity of MILP formulation limits the practicality of the solution when the problem

scales up. In practical scenarios the incoming traffic demand and network size can scale, thus, an efficient and adaptable solution is required for solving (4).

Algorithm 1 SCA based Iterative Algorithm

1: **Initialization:** Set $k \leftarrow 0$, initialize feasible values for $\mathbf{X}^{(\mathbf{0})}(t)$ and $\mathbf{Y}^{(\mathbf{0})}(t)$;
2: **repeat**
3: Solve (8) to obtain $\mathbf{X}(t)$, $\mathbf{Y}(t)$ and $\Xi(t)$;
4: Set $k \leftarrow k+1$;
5: $\mathbf{X}^{(k)}(t) \leftarrow \mathbf{X}(t)$;
6: $\mathbf{Y}^{(k)}(t) \leftarrow \mathbf{Y}(t)$;
7: Update $\Xi^{(k)}(t) \leftarrow \Xi(t)$;
8: **until** Convergence or $(|\Xi^{(k)}(t) - \Xi^{(k-1)}(t)| \leq \epsilon)$;
9: **Output:** $\mathbf{X}(t)$, $\mathbf{Y}(t)$;

B. Proposed Iterative Algorithm to Solve (4)

We propose a SCA based iterative algorithm to solve the problem defined in (4) the pseudo-code of which is presented in Algorithm 1. The key idea behind the proposed algorithm is to relax binary constraints on $\mathbf{X}$ and $\mathbf{Y}$, and introduce following penalty function to accelerate the convergence of iterative algorithm:

$$\mathcal{Z}(\mathbf{X}(t),\mathbf{Y}(t)) = \sum_{p\in\mathcal{P}(t)}\Big(\sum_{i\in\mathcal{R}^E(t)}[(x_{i,p}(t))^2 - x_{i,p}(t)] + \sum_{l\in t}\sum_{j\in\mathcal{R}^M(t)}[(y_{j,p,l}(t))^2 - y_{j,p,l}(t)]\Big). \tag{5}$$

It is clear that for any value of $x_{i,p}(t)$, $y_{j,p,l}(t) \in [0,1]$ the value of $\mathcal{Z}(\mathbf{X}(t),\mathbf{Y}(t))$ is either 0 or negative which is useful to obtain exact binary solutions for the relaxed variables $\mathbf{X}(t)$ and $\mathbf{Y}(t)$. Thus, the problem (4) can be approximated as follows:

$$\begin{aligned}&\max_{\mathbf{X},\mathbf{Y}}: \ OJ + \omega\mathcal{Z}(\mathbf{X}(t),\mathbf{Y}(t)) \qquad (6)\\ &\text{s.t: C3 - C7}\\ &0 \leq x_{i,p}(t) \leq 1, \ \forall\, i \in \mathcal{R}^E(t), p \in \mathcal{P}(t)\\ &0 \leq y_{j,p,l}(t) \leq 1, \ \forall\, l \in t, j \in \mathcal{R}_l^M(t), p \in \mathcal{P}(t),\end{aligned}$$

where OJ is the original objective function in (4) and ω is the weight of the penalty function.

Although all the constraints of the relaxed problem (6) are linear, solving the relaxed problem is still challenging because of the convexity of the the penalty function. To tackle this difficulty, we employ the first-order approximation and linearize the penalty function. Let $x_{i,p}^{(k)}, y_{j,p,l}^{(k)}$ be feasible values at the k-th iteration, in the $(k+1)$-th iteration, the penalty function can be approximated as:

$$\mathcal{Z}^{(k)}(\mathbf{X}(t),\mathbf{Y}(t)) = \sum_{\mathcal{P}}\Big[\sum_{\mathcal{R}^E(t)}\Big(x_{i,p}(t)(2x_{i,p}^{(k)}(t)-1) - (x_{i,p}^{(k)}(t))^2\Big) + \sum_{l\in t}\sum_{\mathcal{R}_l^M(t)}\Big(y_{j,p,l}(t)(2y_{j,p,l}^{(k)}(t)-1) - (y_{j,p,l}^{(k)}(t))^2\Big)\Big]. \tag{7}$$

Finally, the convex approximate of optimization problem in (6) at the $(k+1)^{(th)}$ iteration can be stated as:

$$\begin{aligned}&\max_{\mathbf{X},\mathbf{Y}}: \ \Xi^{(k)}(t) \triangleq OJ + \omega\mathcal{Z}^{(k)}(\mathbf{X}(t),\mathbf{Y}(t)) \qquad (8)\\ &\text{s.t: C3 - C7.}\\ &0 \leq x_{i,p}(t) \leq 1, \ \forall\, i \in \mathcal{R}^E(t), p \in \mathcal{P}(t).\\ &0 \leq y_{j,p,l}(t) \leq 1, \ \forall\, l \in t, j \in \mathcal{R}_l^M(t), p \in \mathcal{P}(t).\end{aligned}$$

The Algorithm 1 provides the summary of proposed SCA-based iterative algorithm. We first initialize feasible points for $\mathbf{X}^{(0)}(t)$ and $\mathbf{Y}^{(0)}(t)$, and then solve (8) for those points. The obtained solution from the initial iteration is then fed as input value of $\mathbf{X}^{(k)}(t)$ and $\mathbf{Y}^{(k)}(t)$ for the next iteration and the steps 3-7 are repeated until the algorithm converges or the difference between the current and previous objective function value is less than or equal to ϵ.

C. Initialization and Convergence of Iterative Algorithm

1) Initialization: The Algorithm 1 works iteratively so we need to initialize values for inner approximation of (8). The feasibility of solution is at first determined by the initial values. For that we randomly assign values between 0-1 for each path through which the request i and j could be routed. The random values are then normalized in such a way that for each request the sum of all paths is exactly equal to 1. The resulting values are then used as the initial feasible points.

2) Convergence: The Algorithm 1 produces a series of improved results for $\mathbf{X}^{(k)}(t)$ and $\mathbf{Y}^{(k)}(t)$ with each iteration. Moreover, there is an increase in the value of objective function $\Xi^{(k)}(t)$ from the initial value. The algorithm stops when it reaches convergence defined by negligible increase in the objective function, i.e., $|\Xi^{(k)}(t) - \Xi^{(k-1)}(t)| \leq \epsilon$, where ϵ is a small constant value. It is shown in simulation results that the proposed iterative algorithm converges in less than 10 iterations depicting the fast convergence.

IV. Results Evaluation

A. Simulation Setup

We consider an integrated LEO-Terrestrial topology consisting of 6 ground gateways, and 30 LEO satellites. Each time slot comprises of 20 sec interval and is further divided into 20 sub-slots of 1 sec each. The connectivity between LEO-LEO, and LEO-gateway changes at the beginning of different time slots. The LEO satellites operate at an altitude of 800 km and are divided into 5 orbits with 6 satellites each. The available link bandwidth for ISLs is set to 75 MHz, while the link bandwidth for GSLs is considered to be 100 MHz, we assume a spectral efficiency of 1 b/s/Hz. The arrival of eMBB and mMTC requests follows Poisson distribution with a mean of λ_e and λ_m requests per time slot, respectively. The minimum capacity for each eMBB request is considered to

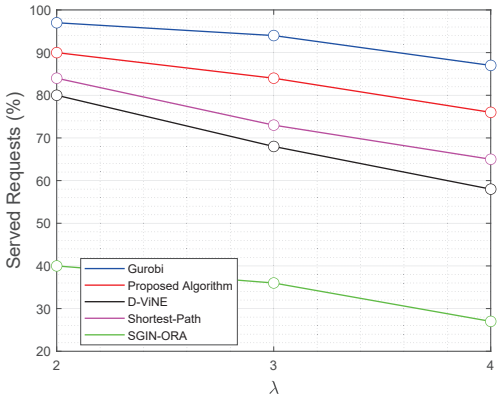

Fig. 2: Percentage of served requests w.r.t total arrival rate.

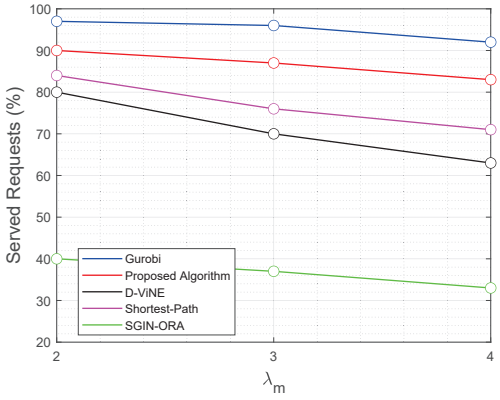

Fig. 3: Percentage of served requests w.r.t mMTC arrival rate.

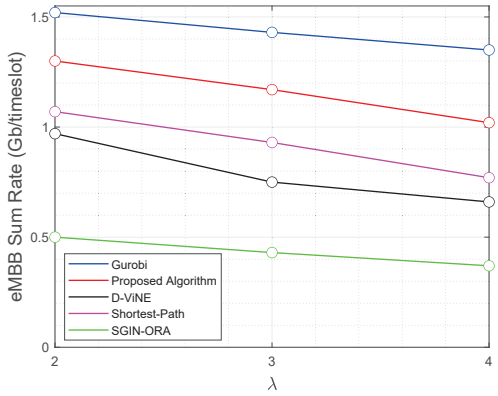

Fig. 4: average eMBB sum rate w.r.t total arrival rate.

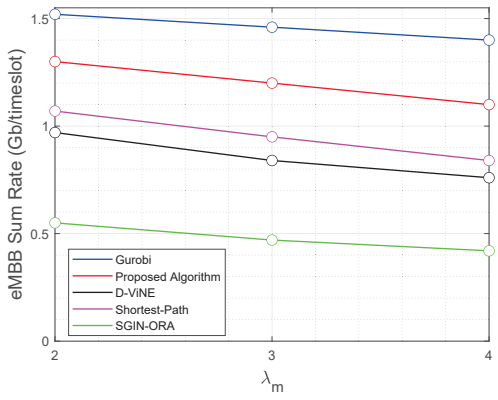

Fig. 5: average eMBB sum rate w.r.t mMTC arrival rate.

be 50 Mbps, whereas maximum tolerable delay for mMTC requests is considered to be 20 ms. The request size of each eMBB and mMTC request is considered to be 2 Gb, and 1 Mb, respectively.

B. Benchmark Schemes

We compare the performance of proposed scheme with 3 benchmark algorithms that include the shortest path, D-VINE from [15], and SGIN-ORA in [11]. In the shortest path algorithm each request is routed through the shortest path available between the source and destination, and the requests are served on the first come first serve basis. The D-VINE approach in [15] involves the routing and resource allocation for VNRs in a time varying satellite topology with the information of current and next time slot. They solved the mathematical model using binary relaxation algorithm, and considered only one type of traffic but we extended the model to include multiple type of requests. The requests arrive at the start of each time slot and occupy the link resources for the entire duration of time slot. The SGIN-ORA approach proposed in [11] involves the routing and resource allocation for diverse slices where a priority is assigned to each slice. They divided the problem into two sub-problems, proposed a heuristic algorithm to determine slice admittance, and solved the linearized optimization model to determine the resources allocated to each slice.

C. Results and Discussion

The key performance metrics used in this article include the percentage of served requests and the average eMBB sum rate. The Fig. 2 and 3 depict the results for the percentage of served requests with the variation in the arrival rate. In Fig. 2 the results are obtained for the case where λ_e and λ_m is increased at the same time represented by λ for the sake of simplicity, while in Fig. 3 only λ_m is increased whereas λ_e is fixed to 2. In both cases, the percentage of served requests decrease with the increase in arrival rate. The results obtained by implementing the proposed iterative algorithm show upto 11% and 18% improvement compared to the shortest path and D-Vine schemes, whereas a massive difference is observed in comparison to SGIN-ORA. The reason for that is in shortest path and D-Vine schemes the mMTC requests occupy the resources for the full duration of a time slot, so a path can accommodate a lesser number of requests during each time slot. This results in lesser available capacity for the incoming requests compared to the proposed scheme where the mMTC requests occupy link capacity for sub-slots, ultimately leading to greater number of served requests. The SGIN-ORA scheme employs priority based resource provisioning where each request occupies link resources for the entire duration of time slot resulting in significant performance degradation compared to the proposed scheme. The iterative algorithm provides significantly improved performance compared to the

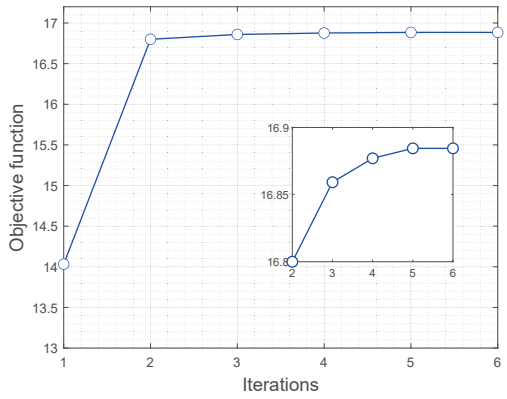

Fig. 6: Convergence behaviour of iterative algorithm with $\lambda = 4$.

benchmark schemes but as expected, it is not able to compete with the optimal solution obtained through solver.

The Fig. 4 and Fig. 5 describe the results obtained in terms of average eMBB sum rate with an increase in the request arrival rate. The sum rate is the average rate available for each eMBB request on a path. As shown in Fig. 4 with the increase in the arrival rate of both eMBB and mMTC requests shown by λ, the eMBB sum rate decreases as more capacity is allocated to the mMTC requests and higher number of eMBB requests arrive during each time slot. The proposed technique outperforms the benchmark schemes as the capacity occupied by mMTC requests is made available for eMBB requests after being served in the sub-slots resulting in higher overall eMBB sum rate in each time slot. Similarly, In Fig. 5 when λ_m is increased while λ_e is fixed to 2, the eMBB sum rate decreases but a slightly higher sum rate is achieved compared to the previous case. The reason is that the average eMBB sum rate not only depends on the capacity allocated to the mMTC requests but also on the number of eMBB requests passing through the path.

Finally, the Fig. 6 shows the convergence behaviour of the proposed iterative algorithm at an arrival rate of $\lambda = 4$. As shown in Fig. 6 the value of objective function increases with each iteration and the algorithm converges in less than 10 iterations showing fast convergence as mentioned in the previous section.

V. Conclusion

In this article, we presented a flexible resource allocation framework to cater the eMBB and mMTC requests in a time-varying LEO satellite topology. We introduced a new sub-slot concept to efficiently allocate the resources to mMTC requests, which are more puncture compared with eMBB services. The eMBB requests arrive at the beginning of time slot and occupy link resources for the entire duration of one or multiple time slots. The mMTC requests arrive at the start of sub-slots within a time slot and are served in few sub-slots. The sub-slot concept allow to release the capacity occupied by the mMTC request after its expired sub-slot. We formulated a mathematical model based of MILP with the objective to maximize the percentage of served requests. To convexify the model and make it computationally tractable we applied binary relaxation technique and solved it through iterative algorithm based on SCA. The simulation results depict the proposed scheme can adapt to variations in the incoming traffic and provide improved performance in terms of percentage of served requests and average eMBB sum rate compared to the benchmark schemes.

Acknowledgment

This work was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant references IPBG19/14016225 and C22/IS/17220888. For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the authors have applied a Creative Commons Attribution 4.0 (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

References

[1] W. Jiang *et al.*, "The road towards 6G: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.

[2] T. K. Rodrigues *et al.*, "Network slicing with centralized and distributed reinforcement learning for combined satellite/ground networks in a 6G environment," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 104–110, 2022.

[3] X. Lin *et al.*, "On the path to 6G: Embracing the next wave of low earth orbit satellite access," *IEEE Communications Magazine*, vol. 59, no. 12, pp. 36–42, 2021.

[4] H. Nguyen-Kha *et al.*, "LEO-to-user assignment and resource allocation for uplink transmit power minimization," in *WSA & SCC 2023; 26th International ITG Workshop on Smart Antennas and 13th Conference on Systems, Communications, and Coding*, 2023, pp. 1–6.

[5] G. Interdonato *et al.*, "On the coexistence of eMBB and URLLC in multi-cell massive MIMO," *IEEE Open Journal of the Communications Society*, 2023.

[6] S. R. Pokhrel *et al.*, "Towards enabling critical mMTC: A review of URLLC within mMTC," *IEEE Access*, vol. 8, pp. 131 796–131 813, 2020.

[7] M. Ahsan *et al.*, "Efficient network slicing for 5G services in Cloud Fog-RAN Deployment Over WDM network," *IEEE Transactions on Vehicular Technology*, 2023.

[8] H. H. Esmat *et al.*, "Towards resilient network slicing for satellite-terrestrial edge computing IoT," *IEEE Internet of Things Journal*, 2023.

[9] J. Tao *et al.*, "Time-varying graph model for leo satellite network routing," in *2022 9th International Conference on Dependable Systems and Their Applications (DSA)*, 2022, pp. 486–491.

[10] G. Cui *et al.*, "Latency optimization for hybrid geo–leo satellite-assisted iot networks," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6286–6297, 2023.

[11] A. Kak *et al.*, "Towards automatic network slicing for the internet of space things," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 392–412, 2022.

[12] P. Zhang *et al.*, "Space-air-ground integrated network resource allocation based on service function chain," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7730–7738, 2022.

[13] M. Minardi *et al.*, "Virtual network embedding for NGSO systems: Algorithmic solution and SDN-testbed validation," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022.

[14] Z. Han *et al.*, "Time-varying topology model for dynamic routing in LEO satellite constellation networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3440–3454, 2023.

[15] I. Maity *et al.*, "D-vine: Dynamic virtual network embedding in non-terrestrial networks," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 166–171.

[16] T. Kim *et al.*, "Satellite edge computing architecture and network slice scheduling for IoT support," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 938–14 951, 2022.