Selective Handover and SLA-Aware Resource Allocation for Diverse Services in a LEO Satellite Network

Muhammad Ahsan†, Thang X. Vu†, Ilora Maity†, Mario Minardi‡, and Symeon Chatzinotas†
†Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg
Email:{muhammad.ahsan, thang.vu, ilora.maity, symeon.chatzinotas} @uni.lu
‡SES Telecom S.A. Betzdorf, Luxembourg
Email: mario.minardi@ses.com

Abstract*—The integrated satellite-terrestrial networks (STNs) aim to provide ubiquitous connectivity and support various services with diverse requirements. Each service request has to go through a sequence of virtual network functions (VNFs) that should be mapped on its routing path. The STNs are equipped with limited communication and computation resources, making it challenging to enable heterogeneous services. Furthermore, the movement of satellites causes frequent changes in topology, which can impact the continuity of long-lasting requests. For requests lasting multiple time frames, the VNF mapping update and path recomputation at the beginning of each time frame is computationally expensive and can cause unwanted service interruptions. Therefore, we propose a selective handover strategy where the path recomputation and VNF remapping are done only if there is a change in the previous routing path. The selective handover strategy ensures that only critical handovers are carried out while discouraging unnecessary reconfigurations, which result in service discontinuity. We develop a software-defined networking (SDN) based experimental testbed that allows us to realistically consider the system constraints. The VNF mapping and path computation for a request are done in a proactive manner, and the rate meters are installed on the switches according to the current network traffic to efficiently utilize the available bandwidth. The simulation results certify that the proposed strategy reduces the packet loss by up to 11.5% and 18.5% as compared to the benchmark schemes and provides stable throughput for eMBB services with minimal service-level agreement (SLA) violations, while also ensuring the latency requirements of mMTC.

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

I. Introduction

The satellite-terrestrial networks (STNs) will play a pivotal role in providing global connectivity and support broadband services requiring high data rates and low latency [1]. The STNs consist of satellites operating in different orbits in the space segment and the user devices and gateways in the ground segment. Low Earth orbit (LEO) satellites operate at altitudes of 350–2000 km, offer reduced round-trip latency and can support service-level agreement (SLA) guarantees, including latency and throughput constraints, for diverse service types [2]. These services are generally categorized into three main categories, including ultra-reliable and low-latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC) [3]. The URLLC services have a stringent latency requirement of < 1 ms, which cannot be satisfied through the STNs as the round-trip delay between the LEO satellites and ground gateways or user devices is at least $10 - 20$ ms [4]. Therefore, we focus on the other two broad service categories, eMBB and mMTC. The eMBB services are bandwidth-intensive and are generally long-lasting, while the mMTC requests are bursty in nature, transmitting small data packets at different intervals, often having a constraint for end-to-end (e2e) delay [5].

The LEO satellites are capable of supporting eMBB and mMTC use cases but a major challenge arises due to the limitation of onboard computation and communication resources as each request has to go through a sequence of virtual network functions (VNFs), i.e. compression, decompression, load balancing, network address translation (NAT) etc, that should be executed on the nodes constituting its routing path as shown in Fig. 1. In addition, the mobility of satellites causes frequent changes in satellite topology, which necessitate handovers. Frequent handovers can be very costly as they disrupt the data flow for ongoing service requests, causing SLA violations. Therefore, we propose a selective handover strategy where the VNF mapping and the routing path are only updated in the subsequent configuration period (used interchangeably with time frame) if the previous routing path is no longer available. In this way, the selective handover strategy avoids unnecessary reconfigurations while enabling essential handovers to maintain service continuity.

Related Works: In recent years, various studies in the literature focused on routing and resource allocation in satellite networks [6]–[8]. In [9], the authors introduced a dynamic inter-satellite path computation approach where the path between two satellites was selected using a time-space graph built from link attributes over multiple time steps. Literature [10] proposed a routing and resource allocation strategy over a cubeSAT network with the aim of minimizing the SLA violations. They proposed a heuristic algorithm for route computation and addressed resource allocation by linearizing the original mixed-integer nonlinear programming (MINLP) formulation. Mario *et al.* in [11] implemented an SDN-based

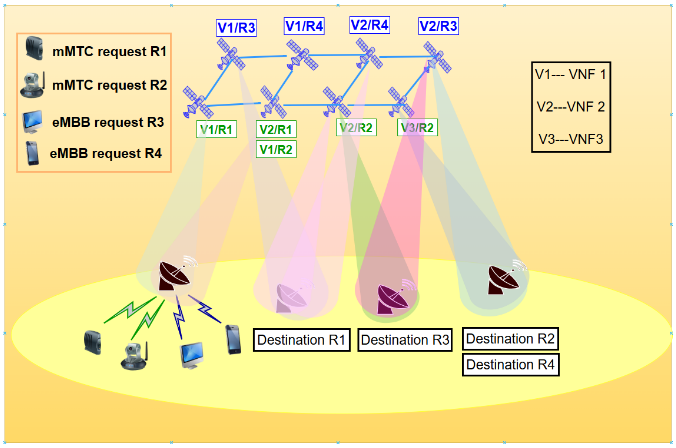

Fig. 1: The system model.

experimental testbed for routing of virtual network requests (VNRs) during different time slots. In [12], the authors presented a deep reinforcement learning (DRL) based framework for joint path selection and bandwidth allocation for multiple network slices in the STNs. To address the dynamic behavior of the satellite topology, the time frame was divided into multiple time slots; within a time slot, the network topology was assumed to remain quasi-static. The authors employed a graph attention network (GAT) to collect the network status during each time slot, which was then used by the DRL agent for path selection and subsequent bandwidth allocation.

More recently, research efforts have extended these approaches to the joint optimization of VNF placement, routing, and resource allocation [13]–[15]. Literature [16] presents a DRL–based approach for virtual network embedding (VNE), in which embedding decisions are guided by the real-time availability of substrate network resources. Abreha *et al.* in [17] proposed a VNF placement strategy where the path and VNF configuration took place at the start of each time slot. The authors modeled the VNF placement problem as an integer nonlinear programming (INLP) formulation and solved it using heuristic algorithms. The authors in [18] proposed an optimization scheme for VNF placement over a LEO satellite network. The VNF placement was carried out proactively at the start of a configuration period with known request arrival time, and all the requests were provisioned within a configuration period. They linearized the original INLP to an integer linear programming (ILP) problem and divided it into multiple sub-problems to solve them iteratively. In addition, the authors proposed a greedy-based heuristic algorithm to reduce the time complexity of the solution. Literature [19] proposed a VNF mapping and scheduling scheme where the VNF placement was updated dynamically to cater to the latency requirements of the previous and new arriving requests. They solved the problem using a Tabu search-based algorithm.

Motivation and Contributions: It is clear from the review of previous literature that the existing studies lack the planning for requests lasting for more than a time frame and try to cater to them within the time frame or reconfigure too often, which can have a negative impact on the continuity of ongoing requests. In addition, most existing studies primarily concentrate on optimization frameworks and machine learning–based solutions, with comparatively limited emphasis on testbed-based implementations. Therefore, our main contributions can be summarized as follows.

- We introduce a selective handover strategy for multi-frame requests with the aim of reducing SLA violations and packet loss due to frequent handovers. The VNF mapping update and path recomputation for ongoing requests only occur if the previous routing path is not available. Moreover, for ongoing requests not requiring a handover, resources are reserved in links and nodes of the path to prioritize them over the new incoming requests.
- We design an experimental testbed in which an SDN-based control plane dynamically orchestrates network resources using a selective handover and dynamic bandwidth allocation (SHDBA) algorithm. The algorithm jointly performs VNF mapping, routing, and resource allocation for incoming requests while taking into account the SLA guarantees in terms of completion time for mMTC and achieved throughput for eMBB services.
- We validate the SHDBA algorithm on an SDN-enabled Mininet-based testbed with real traffic generation and compare its performance with baseline schemes in terms of average packet loss and SLA violation rate. In addition, we analyze the performance of SHDBA in terms of eMBB throughput and mMTC completion time for different requests with varying SLA guarantees.

The rest of the article is organized as follows. In Section II, we present the proposed SHDBA algorithm along with the network and service model. In Section III we describe the working and setup of the testbed. The results are evaluated in Section IV. Finally, we will conclude the paper in Section V.

II. System Network Model and Proposed Solution

A. Network and Service Model

We consider a network consisting of multiple users on the ground and LEO satellites operating in the space segment. The users are divided into two main categories, the eMBB and mMTC. Each eMBB user request is characterized by its source, destination, minimum required rate R_r^{min}, required VNFs, and the duration. The mMTC requests are characterized by the source, destination, maximum tolerable delay d_r^{max}, size s_r, and the required VNFs. The minimum required rate is then calculated as the ratio of the request size s_r and d_r^{max}. We employ a proactive routing and resource allocation approach considering the predictable movement of the satellites. A time frame is divided into multiple time slots, with the VNF mapping considered fixed for each request during a time frame, while the topology is considered quasi-static within a time slot. For eMBB requests, the path computation is done at the start of the time frame, as they require resources for a longer time. Similarly, the paths for mMTC requests are also computed at the start of a time frame, but the resources are allocated on that path during the time slot where the mMTC user transmits. This reduces the path computation delay for the mMTC users, which is critical considering their latency constraint. For eMBB users, the choice of the path is motivated by its availability for possibly the whole time frame.

Algorithm 1 Selective Handover and Dynamic Bandwidth Allocation Algorithm (SHDBA)

Inputs: Set of all the links $\rightarrow \mathcal{L}$, new arriving requests $\rightarrow \mathcal{R}_N$, continuing requests $\rightarrow \mathcal{R}_C$, minimum rate requirement of a request $\rightarrow R_r^{min}$, required VNFs for each request, link capacities $\rightarrow c_l$,
Execution:
Start of Time frame
if there is a topology change **then**
    $\mathcal{R}_H \leftarrow$ affected requests from set $\mathcal{R}_C$ requiring
    handover;
end if
for $r \in \mathcal{R}_N \cup \mathcal{R}_H$ **do**
    $\rightarrow$ Compute least cost path for r (Dijkstra Algorithm);
    $\rightarrow$ Choose a path that can host the required VNFs;
end for
for $l \in \mathcal{L}$ **do**
    $\rightarrow$ Ensure $c_l > \mathcal{R}_r^{min}$ for all the requests sharing l,
    while R_r^{min} for $r \in \mathcal{R}_C/\mathcal{R}_H$ is reserved;
    $\rightarrow$ Divide the residual capacity among the requests;
    $\rightarrow$ Store the available capacity per link for r in $\mathcal{C}_r$;
end for
for $r \in \mathcal{R}_N \cup \mathcal{R}_C$ **do**
    Configure HTB queues with R_r^{min} on each node;
    Install a meter on each node for $R_r^{max} = min(\mathcal{C}_r)$;
end for
Output: Flow rules and rate meters installation for each request.

B. Proposed SHDBA Algorithm

*1) **Objective**:* The objective of the SHDBA algorithm is to maximize the served requests while ensuring their SLA requirements.

*2) **Constraints**:* The SHDBA considers the following constraints.

- Each request should be mapped onto one path.
- The nodes of the chosen path should have enough computation capacity to execute all the VNFs for a request.
- Rate of a request should be greater than R_r^{min}.
- Rate of all the requests routed through paths using a link should be less than its capacity.
- Completion time of mMTC requests should be less than the maximum allowable delay indicated in the SLA guarantee.

*3) **Steps of SHDBA**:* The steps of the SHDBA algorithm are provided in the pseudo-code given as Algorithm 1. At the start of a time frame, the algorithm checks the changes in topology as compared to the previous time frame and puts the requests that are affected by the change in topology in set $\mathcal{R}_H$. The R_r^{min} and resources required for VNF execution are reserved in the links and nodes of the path for a request not requiring a handover. After this, the algorithm computes the least cost feasible path using the Dijkstra algorithm [20], for each new arriving request and for those requiring a handover. The cost of the path is determined based on its stability for eMBB requests and its propagation delay for mMTC. The chosen path should have sufficient computation capacity to execute all the required VNFs for a request. Moreover, the algorithm ensures that each link has enough capacity to at least provide R_r^{min} for the requests passing through it. If there is residual capacity left in a link, then it is divided equally among all the requests passing through it. The achievable rate on each link constituting the routing path for a request is stored in a set $\mathcal{C}_r$. Finally, the algorithm forwards the rate meters to be installed on the nodes of the routing path to the SDN controller with a minimum rate R_r^{min} and maximum achievable rate R_r^{max}, where R_r^{max} is the minimum rate on any link along the routing path.

C. Distinction from Existing Solutions

As described earlier in Section I, routing, resource allocation, VNF mapping and migration in the satellite networks have been the focus of various studies in the literature. The solution approaches primarily adopted in previous works involve optimization and heuristic-based solutions where the routing path computation and VNF mapping are done while considering the constraints for available link bandwidth and computation resources at the nodes [17]–[19]. In [17], the authors proposed a VNF mapping and scheduling strategy for mission-critical applications where the network reconfigurations were carried out at the start of each time slot to plan for the current and the next time slot. They formulated the problem as an INLP with the objective of minimizing the e2e delay and solved it using heuristic algorithms based on simulated annealing. Literature [18] proposed an optimization framework for joint routing and VNF placement with the objective to minimize the e2e latency of the service requests. They divided the network configuration period into time slots, with all the requests starting at the beginning of the configuration period and served within its duration. The authors in [19] proposed a VNF mapping and migration strategy where the mapping and scheduling decisions were updated upon the arrival of each new request to satisfy the latency requirements of incoming and existing requests. In [16], the authors proposed a DRL-based approach for VNF mapping and resource allocation where the RL agent is trained to make the VNF and path mapping decision based on the available resources and network topology during each time slot.

Mario *et al.* in [21] implemented a time slot-based path mapping strategy in a Mininet-SDN-based testbed with an objective to provide the different classes of VNRs with a minimum rate to satisfy their User Satisfaction Probability (USP) according to their priority. The path mapping and

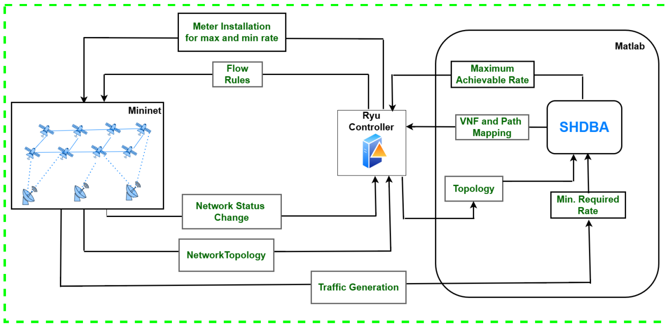

Fig. 2: Block diagram of the testbed.

the resource update were done in every time slot with the bandwidth of expired VNRs released for the use of new arriving requests. Similarly, the authors in [11] proposed a path mapping strategy with the aim of minimizing the migration cost due to a change in topology in different time slots. They implemented the problem in a Mininet-SDN-based testbed, with the path for an ongoing VNR recomputed at the start of each time slot based on the current network topology.

In our proposed solution, unlike all the previous works, we plan for eMBB requests lasting for more than one time frame, using the selective handover strategy that prioritizes ongoing requests not requiring a handover through resource reservation. A new path and VNF mapping are only updated for an ongoing request if the previous routing path from the last time frame is no longer available. This ensures a continuous data flow without outages for requests not requiring a handover, while also enabling critical handovers for the continuity of requests whose routing path and VNF mapping need to be updated. Moreover, the provisioning of mMTC requests takes place during time slots (where they require data transmission) within a time frame with a constraint on their completion time as they have a stringent requirement in terms of latency. The proposed algorithm is implemented on a Mininet–RYU–MATLAB-based testbed setup, which generates real traffic through iPerf, applies meters on switches according to the SLA guarantees and available resources, ensuring the link capacity constraint, and captures dynamic network behavior, providing a realistic performance assessment.

III. Testbed Setup

The Fig. 2 shows the block diagram of the testbed. The testbed consists of three main components the network emulator Mininet [22], RYU as the SDN controller [23], and MATLAB running the SHDBA algorithm. First, the satellite topology for different time frames is generated offline using satellite communication and aerospace toolbox in MATLAB to have a realistic time varying behavior of the topology. We then emulate the network in Mininet with the SDN-enabled OpenFlow switches representing the satellites and Mininet hosts representing the user devices. The capacity of each link is indicated in the topology setup along with the delay calculated based on the length of inter-satellite (ISL) and ground-to-satellite (GSL) links. The RYU controller is the intermediate entity between the emulated network and the Algorithm 1 executed in MATLAB. The Algorithm 1 gets the topology information from Mininet through the RYU controller, and based on that, computes the routing paths for the requests in each time frame. The Algorithm 1 then computes the maximum achievable rates for the request and the flow rules and the meters for minimum and maximum rates are then installed on the switches by the SDN controller. Once the flow rules and meters are installed, real traffic is generated between Mininet hosts using iPerf to emulate realistic network behavior. The traffic parameters, including data rate and session duration, are configured according to the service requirements of eMBB and mMTC flows. Performance metrics such as throughput, e2e delay, packet loss, and flow completion time are continuously monitored during the experiments. In this way, the setup enables an accurate evaluation of the proposed algorithm under dynamic traffic and varying network conditions.

IV. Results Evaluation

A. Simulation Setup

We consider a satellite network consisting of 16 satellites and 24 users scattered at different locations across Europe. The satellites are divided into 2 orbits operating at an altitude of 780 km with an angle of inclination equal to 86.4^o. The users are connected to the satellites through the GSLs, having a capacity of 100 Mbps, while satellites are connected through ISLs, each having a capacity of 150 Mbps. The duration of a time frame is set to 600 sec and it is divided into four time slots. The eMBB requests arrival follows a Poisson distribution with a mean of λ_e requests per time frame. The mMTC users send data periodically and often send packets every few minutes, so we assume that each mMTC user will transmit once per time frame during a random time slot within a time frame. The SLA requirement in terms of R_r^{min} for the eMBB users is chosen between $10-50$ Mbps. The size of the mMTC requests generated by each user at different time frames is uniformly distributed between $\mathcal{U}(0.5, 1)$ Mb for users $1-4$, $\mathcal{U}(1, 2)$ Mb for users $5-8$, and $\mathcal{U}(2, 3)$ Mb for users $9-12$, having an SLA guarantee for a completion time less than 100, 200, and 300 ms, respectively [24]. The required number of VNFs for each request is randomly distributed between $2-4$ while we assume that the execution of each VNF requires a $10-20$ ms interval, as we employ lightweight container-based VNFs such as compression, decompression, and NAT that are smaller in size and require less execution time.

1) Packet Loss Comparison: We compare the performance of the proposed algorithm in terms of average packet loss per request at different eMBB request arrival rates λ_e, with two algorithms from recent literature, the TEDG algorithm from literature [18], and the FASD-MASC from [17]. In the TEDG algorithm, all the requests arriving during a time frame are catered within its duration, and multi-frame requests are not considered. In FASD-MASC, the VNF and path reconfigurations are done at the start of each time slot, planning only for the current and next time slot. Each instance of the bar plot shown in Fig. 3 is an average of 50 realizations at

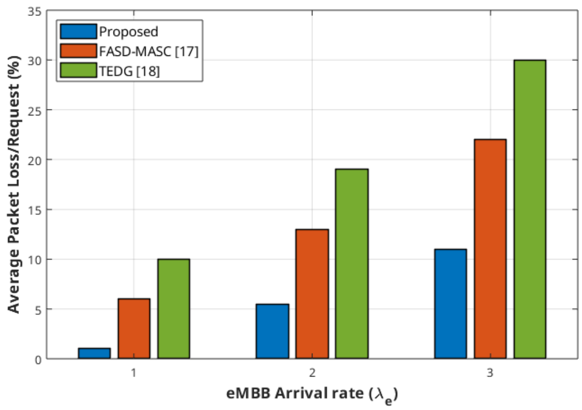

Fig. 3: Average packet loss percentage comparison for eMBB requests.

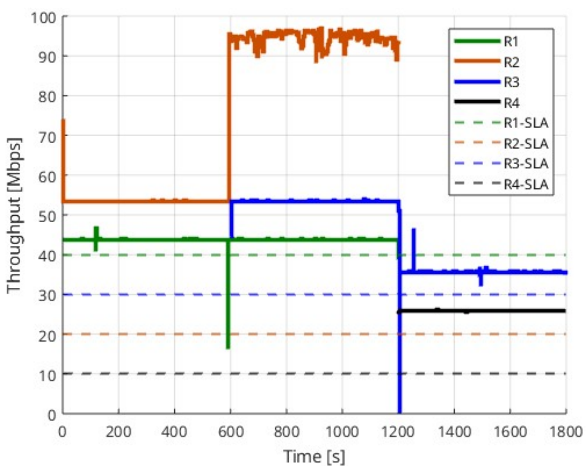

Fig. 4: Achieved throughput for eMBB requests over time.

$\lambda_e = 1, 2$, and 3. As shown in Fig. 3, the proposed strategy significantly reduces average packet loss per request as compared to the TEDG and FASD-MASC, as the proposed scheme only considers reconfiguration for requests whose routing path from the previous time frame is no longer available, thus avoiding the outage due to frequent handovers, which reduces the packet loss. On the other hand, in FASD-MASC, the reconfigurations occur too frequently, which ultimately results in higher packet loss, while in TEDG, the rate requirement for requests increases to provision them within a time frame, which results in a lack of link capacity when the number of requests increases, causing higher packet loss.

2) eMBB Throughput: The Fig. 4 depicts the real-time throughput analysis for eMBB requests with $\lambda_e = 2$ requests per time frame. The SLA guarantee for each request in terms of R_r^{min} is randomly chosen from 10, 20, 30, 40, and 50 Mbps. The data transmission for requests 1 and 2 begins at the start of the first time frame, while the transmission for requests 3 and 4 begins at the start of time frame 2 and 3, respectively, with each time frame comprising 600 sec. As shown in Fig. 4, the request R1 achieves a throughput greater

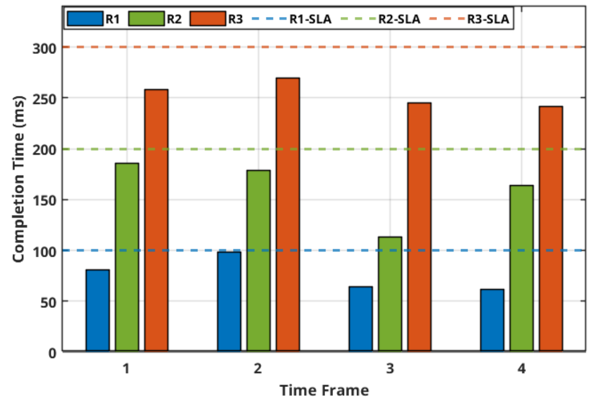

Fig. 5: mMTC requests completion time during different time frames.

Algorithm	$\lambda_e = 1$	$\lambda_e = 2$	$\lambda_e = 3$
FASD-MASC [17]	1.1%	2.5%	4.4%
TEDG [18]	1.4%	3.2%	5.7%
Proposed	0.01%	0.1%	0.25%

TABLE I: Average SLA violation period per eMBB request in terms of throughput at different arrival rates.

than its R_r^{min} for the entire duration of time frame 1, but its throughput drops at the start of time frame 2 as it goes through a handover that results in a sudden drop in throughput. The throughput then stabilizes after a new path is computed for R1 and increases above the R_r^{min} for the remaining duration of the request. The achieved throughput for request R2 does not violate its SLA guarantee at any instant during transmission, as its routing path remains intact for the entire duration, and there are no handover-induced outages. The throughput for request R3 drops below the SLA at the start of time frame 3 due to a handover, but once a new path is computed, it stabilizes above R_r^{min} for its remaining duration. The request R4 started and finished within the time frame 3, thus not requiring a handover and achieving a throughput greater than R_r^{min} for its entire duration. Table I shows a comparison between our proposed algorithm and the baseline schemes in terms of average SLA violation duration per eMBB request at different arrival rates. It is clear that the proposed algorithm significantly reduces the SLA violations even for higher λ_e, as selective handover and resource reservation result in a significant decrease in handover-induced outages.

3) mMTC Completion Time: The Fig. 5 depicts the results in terms of mMTC request completion time during different time frames as compared to their SLA guarantees. As described earlier, the mMTC users send packets periodically after a certain time duration, so we plot their completion time during each time frame. The SLA guarantee for requests R1, R2, and R3 is set to 100, 200, and 300 ms, with their sizes $\mathcal{U}(0.5, 1), \mathcal{U}(1, 2), \mathcal{U}(2, 3)$ Mb, respectively. As shown in Fig. 5, the completion time for each request is within the limit of SLA guarantee as the R_r^{min} is calculated based on the request size and maximum allowable latency. The proposed SHDBA strategy sets the meter for R_r^{min}, and the achieved throughput is more than that, as the residual capacity in the links is divided among the requests passing through it. This ensures that each mMTC request is provided with sufficient resources at each time frame for transmission without any SLA violation.

V. Conclusion

In this article, we presented a selective handover strategy where the VNF remapping and path recomputation for a continuing request from the previous time frame were only done if there was a change in the routing path. We implemented an SDN-based experimental testbed where a satellite network topology was emulated in Mininet and dynamically orchestrated using the SHDBA algorithm. The testbed computes the routing paths based on the current network topology and the incoming demand. Moreover, it dynamically sets the meters on the switches with the minimum and maximum rates for requests based on the SLA guarantees and the available path capacity. The simulation results show that the proposed strategy outperforms the benchmark algorithms in terms of the percentage of packet loss. In addition, the proposed strategy meets the SLA guarantees for eMBB and mMTC users in terms of achieved throughput and request completion time with minimal violations. In future work, we plan to incorporate reactive routing into the existing approach to handle unexpected changes in network topology. Additionally, we aim to further enhance the scalability of the solution to support ultra-dense satellite constellations.

Acknowledgment

This work was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant references IPBG19/14016225, C22/IS/17220888/RUTINE, and DEFENCE22/IS/17812162/SDNSaT. For the purpose of open access, and in fulfillment 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] T. Kim, J. Kwak, and J. P. Choi, "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.

[2] M. Ahsan, T. X. Vu, and S. Chatzinatos, "Flexible resource allocation for embb and mmtc services in a time-varying satellite topology," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2024, pp. 1–6.

[3] Y. Liu, B. Clerckx, and P. Popovski, "Network slicing for embb, urllc, and mmtc: An uplink rate-splitting multiple access approach," *IEEE Transactions on Wireless Communications*, vol. 23, no. 3, pp. 2140–2152, 2024.

[4] International Telecommunication Union Radiocommunication Sector (ITU-R), "Technical and operational aspects of non-terrestrial networks (ntn) in the imt system," ITU Radiocommunication Sector (ITU-R), Geneva, Switzerland, Report M.2514, 2022.

[5] M. Ahsan, A. Ahmed, H. Fida, A. Al-Dweik, U. S. Hashmi, and A. Ahmad, "Efficient network slicing for 5g services in cloud fog-ran deployment over wdm network," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 12 099–12 112, 2023.

[6] M. Ahsan, T. X. Vu, and S. Chatzinotas, "Flexible resource allocation and path reconfiguration strategies for embb and mmtc services in a leo satellite topology," *IEEE Transactions on Vehicular Technology*, pp. 1–12, 2025.

[7] M. Minardi, Y. Drif, T. X. Vu, and S. Chatzinotas, "Sast-vne: A flexible framework for network slicing in 6g integrated satellite-terrestrial networks," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 1, pp. 234–244, 2025.

[8] Y. Wang, L. Zhao, X. Chu, S. Song, Y. Deng, A. Nallanathan, and G. Zhou, "E2e network slicing optimization for control- and user-plane separation-based sagins with drl," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 16 680–16 696, 2024.

[9] Z. Han, C. Xu, G. Zhao, S. Wang, K. Cheng, and S. Yu, "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.

[10] A. Kak and I. F. Akyildiz, "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.

[11] M. Minardi, T. X. Vu, L. Lei, C. Politis, and S. Chatzinotas, "Virtual network embedding for ngso systems: Algorithmic solution and sdn-testbed validation," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3523–3535, 2023.

[12] L. Pan, X. Yang, and Z. Bu, "Intelligent network slicing optimization for satellite-terrestrial integrated networks," *IEEE Communications Letters*, vol. 28, no. 10, pp. 2342–2346, 2024.

[13] I. Maity, T. X. Vu, S. Chatzinotas, and M. Minardi, "D-vine: Dynamic virtual network embedding in non-terrestrial networks," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 166–171.

[14] Y. Yue, X. Tang, W. Yang, X. Zhang, Z. Zhang, C. Gao, and L. Xu, "Delay-aware and resource-efficient vnf placement in 6g non-terrestrial networks," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2023, pp. 1–6.

[15] L. Tang, W. Wen, J. Li, D. Fang, L. Li, and Q. Chen, "Digital-twin-assisted vnf mapping and scheduling in sdn/nfv-enabled industrial iot," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 18 516–18 533, 2024.

[16] I. Maity, T. X. Vu, and S. Chatzinotas, "Revine: Reinforcement learning-based virtual network embedding in satellite-terrestrial networks," *IEEE Transactions on Communications*, pp. 1–1, 2024.

[17] H. G. Abreha, H. Chougrani, I. Maity, Y. Drif, C. Politis, and S. Chatzinotas, "Fairness-aware vnf mapping and scheduling in satellite edge networks for mission-critical applications," *IEEE Transactions on Network and Service Management*, vol. 21, no. 6, pp. 6716–6730, 2024.

[18] S. Yuan, Y. Sun, and M. Peng, "Joint network function placement and routing optimization in dynamic software-defined satellite-terrestrial integrated networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 5, pp. 5172–5186, 2024.

[19] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-aware dynamic sfc mapping and scheduling in sdn/nfv-enabled space–air–ground-integrated networks for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5824–5838, 2022.

[20] M. Noto and H. Sato, "A method for the shortest path search by extended dijkstra algorithm," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, vol. 3. IEEE, 2000, pp. 2316–2320.

[21] M. Minardi, T. X. Vu, I. Maity, C. Politis, and S. Chatzinotas, "Traffic-aware virtual network embedding with joint load balancing and datarate assignment for sdn-based networks," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 4936–4948, 2024.

[22] K. Kaur, J. Singh, and N. S. Ghumman, "Mininet as software defined networking testing platform," in *International conference on communication, computing & systems (ICCCS)*, 2014, pp. 139–42.

[23] A. T. Albu-Salih, "Performance evaluation of ryu controller in software defined networks," *Journal of al-qadisiyah for computer science and mathematics*, vol. 14, no. 1, pp. Page–1, 2022.

[24] C. Song, M. Zhang, Y. Zhan, D. Wang, L. Guan, W. Liu, L. Zhang, and S. Xu, "Hierarchical edge cloud enabling network slicing for 5G optical fronthaul," *Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B60–B70, 2019.