sensors

MDPI

Article

Online Service Function Chain Planning for Satellite–Ground Integrated Networks to Minimize End-to-End (E2E) Delay

Soohyeong Kim [1], Joohan Park [2], Jiseung Youn [1], Seyoung Ahn [1] and Sunghyun Cho [3,*]

[1] Major in Bio-Artificial Intelligence, Department of Computer Science and Engineering, Hanyang University, Ansan 15588, Republic of Korea; dreammusic23@hanyang.ac.kr (S.K.); yjs1104@hanyang.ac.kr (J.Y.); tpdud1014@hanyang.ac.kr (S.A.)

[2] Spatial Wireless Networking Research Section, Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Republic of Korea; 1994pjh@etri.re.kr

[3] Department of Computer Science and Engineering, Hanyang University, Ansan 15588, Republic of Korea

* Correspondence: chopro@hanyang.ac.kr; Tel.: +82-31-400-4757

Abstract: The combination of software-defined networking (SDN) and satellite–ground integrated networks (SGINs) is gaining attention as a key infrastructure for meeting the granular quality-of-service (QoS) demands of next-generation mobile communications. However, due to the unpredictable nature of end-user requests and the limited resource capacity of low Earth orbit (LEO) satellites, improper Virtual Network Function (VNF) deployment can lead to significant increases in end-to-end (E2E) delay. To address this challenge, we propose an online algorithm that jointly deploys VNFs and forms routing paths in an event-driven manner in response to end-user requests. The proposed algorithm selectively deploys only the essential VNFs required for each Service Function Chain (SFC), focusing on minimizing E2E delay—a critical QoS parameter. By defining a minimum-hop region (MHR) based on the geographic coordinates of the routing endpoints, we reduce the search space for candidate base stations, thereby designing paths that minimize propagation delays. VNFs are then deployed along these paths to further reduce E2E delay. Simulations demonstrate that the proposed algorithm closely approximates the global optimum, achieving up to 97% similarity in both E2E delay and CPU power consumption, with an average similarity of approximately 90%.

Keywords: LEO satellite; online algorithm; routing algorithm; satellite–ground integrated network; service function chain

check for updates

Citation: Kim, S.; Park, J.; Youn, J.; Ahn, S.; Cho, S. Online Service Function Chain Planning for Satellite–Ground Integrated Networks to Minimize End-to-End (E2E) Delay. *Sensors* **2024**, *24*, 7286. https://doi.org/10.3390/s24227286

Academic Editor: Kai Yang

Received: 17 October 2024
Revised: 7 November 2024
Accepted: 13 November 2024
Published: 14 November 2024

1. Introduction

Satellite–ground integrated networks (SGINs) are expected to play a key role in next-generation communication systems due to their ability to provide global and seamless services [1–6]. In the service scenarios envisioned for next-generation mobile communications, quality of service (QoS) requirements are becoming more granular and demand higher performance [7]. Moreover, next-generation communication systems aim to extend such requirements to global coverage, ensuring seamless service regardless of location [8,9]. While existing ground networks have made strides in meeting these QoS demands, they still suffer from coverage holes in remote areas such as sparsely populated regions, oceans, and volcanic zones. Recently, low-Earth orbit (LEO) satellites have garnered attention as a potential solution to this challenge, thanks to reductions in both launch and maintenance costs [10]. However, LEO satellites have limited computational capabilities, which may hinder their ability to meet the stringent QoS requirements of next-generation service scenarios [11]. Therefore, by combining the strengths of ground and satellite networks, SGIN can effectively address QoS demands while eliminating coverage gaps, positioning itself as a crucial paradigm for future communication systems.

Despite the emergence of SGIN, there are still unresolved challenges in providing next-generation services through SGIN. The hardware of existing LEO satellites is specialized for specific tasks [12,13]. In other words, each satellite can only perform predetermined tasks, leading to low resource utilization when these tasks are not requested. Upgrading hardware or adding new tasks requires launching additional satellites, resulting in high operating costs. Furthermore, integrating LEO networks with terrestrial networks is complex, as the technologies for each have evolved independently, necessitating new interoperability solutions [14].

Applying software-defined networking (SDN) and network function virtualization (NFV) technologies to SGIN offers a solution to these challenges [15]. SDN-based SGIN can enable more efficient service delivery through the use of Service Function Chains (SFCs) and Virtual Network Functions (VNFs). An SFC is an ordered sequence of network services that data traffic must traverse to meet specific requirements. These services are implemented as VNFs, which are virtualized network functions designed to provide flexible and efficient deployment. For example, an SFC might include a firewall (VNF_1), followed by a load balancer (VNF_2) and then an intrusion detection system (VNF_3). This structure allows the network to dynamically manage and optimize service delivery, reducing dependency on specialized hardware and improving flexibility and efficiency [12–14]. Consequently, SDN-based SGIN networks can deploy VNFs on demand at specific nodes, eliminating the need for additional hardware upgrades and enhancing resource utilization, ultimately enabling the provision of next-generation services without significant hardware investment.

1.1. Comprehensive Review of Recent Works

Various studies have been conducted on leveraging SDN in SGIN, particularly focusing on VNF deployment, SDN task offloading, resource allocation, orchestration, and multicast.

Numerous studies have been conducted on VNF placement with various objectives, ranging from minimizing end-to-end (E2E) delay to maximizing resource utilization. In SDN-based LEO networks, a VNF orchestration algorithm based on a time-evolving graph that considers the limitations of LEO satellites' energy and computing resources was proposed in [13]. The authors of [16] suggested a distributed VNF deployment algorithm to minimize the E2E delay and transmission costs while deploying VNFs on LEO edge cloud satellites. A provisioning technique for dynamically determining the configuration of controllers based on control overhead and network load in SDN-based LEO satellite networks has been proposed [17]. The authors of [18] proposed a graph-attention-network-based algorithm for the smooth orchestration of Service Function Chains (SFCs) in LEO satellite networks for load balancing and QoS improvement. The authors of [19] presented an online VNF deployment algorithm based on deep reinforcement learning to maximize the resource utilization of LEO satellites for granular QoS and SFC requirements. In [20], an online algorithm was introduced, which chains VNFs in response to service requests from end users in an on-demand manner to minimize delays in LEO edge computing environments. The authors of [21] proposed a heuristic greedy algorithm to evenly maintain resource consumption among base stations in SGIN, deploy VNFs, and form routing paths for servicing SFCs. The authors of [22] proposed a greedy optimization algorithm to improve the link resource utilization of LEO satellites in SGIN by deploying VNFs and forming routing paths. To provide Internet-of-Vehicle services in SGIN combined with SDN, a Tabu search-based online algorithm for VNF deployment and scheduling to find sub-optimal results was proposed in [23]. The authors of [24] introduced a VNF deployment algorithm based on the stateless architecture of the core network and reliable context management to reduce the transmission frequency and service interruptions in SGIN. The authors of [12] proposed a time-expansion-based decoupled greedy algorithm to decrease the E2E delay while deploying VNFs and forming routing paths in SGIN.

Many studies have addressed topics in SDN-based SGIN, including task offloading and resource allocation. An algorithm for computational task allocation in LEO satellite edge computing was proposed to minimize the energy consumption of ground end users while satisfying QoS requirements [25]. A greedy task allocation algorithm for LEO satellite edge computing nodes was proposed to maximize computing resource utilization and minimize task-uploading costs [26]. The authors of [27] proposed a computation and communication resource allocation algorithm that enables task offloading in SGIN, thereby reducing delays while ensuring LEO satellites are energy-efficient. In [28], a channel-aware gradient fair association algorithm was proposed for collaborative offloading to achieve load balancing between LEO satellites. The authors of [29] designed an architecture for LEO edge-computing satellites supporting IoT devices and proposed a low-complexity offloading and scheduling algorithm. In [30], an online algorithm was introduced for resource allocation and offloading decisions to minimize power consumption for both end users and LEO satellites. In [31], a system model based on the Stackelberg game was designed for situations involving LEO satellite networks and large-scale end users, along with an offloading decision algorithm for end users based on the mean-field game.

Research has also been conducted on orchestration in SDN-based SGIN. In [32], three heuristic approaches were proposed for deploying VNFs on LEO satellites to improve the orchestration between SFCs. The authors of [33] proposed a potential game-based SFC delivery scheme to reduce orchestration and E2E delay among multiple SFCs installed in SGIN, thus optimizing SFC delivery. The authors of [34] proposed utilizing federated learning in SGIN to dynamically allocate resources to SFCs, satisfying the QoS requirements of end users. In [35], a reconfigurable SGIN architecture was designed, and resource allocation and orchestration algorithms were proposed for servicing SFCs.

There are a few studies related to multicast in SDN-based SGIN networks. The works of [36–38] proposed a software-defined multicast framework for improving Internet video delivery performance in SDN-based LEO satellite networks. The authors of [39] also proposed a routing algorithm that utilized unicast and multicast traffic engineering to enhance the performance of inter-satellite links in SDN-based LEO satellite networks.

1.2. Research Gap and Motivation

To date, various studies leveraging SDN in SGIN have been conducted, focusing on areas such as VNF placement, task offloading, and SFC orchestration. Among these, considerable attention has been given to VNF placement strategies to minimize E2E delay. However, two major technical challenges remain unresolved: (i) implementing VNF placement in an event-driven manner and (ii) jointly determining both VNF placement and routing paths. Firstly, service providers cannot predict when or what type of SFC requests will be made by end users [40]. One possible solution is to pre-deploy as many VNFs as possible on SGIN nodes to cope with this unpredictability; nevertheless, the limited storage capacity of satellites limits the number of VNFs that can be deployed. Even if a large number of VNFs were pre-deployed, the high traffic loads passing through these VNFs could quickly saturate the storage and computational capacity, resulting in a significant increase in E2E delay [41]. Therefore, a mechanism is needed to enable service providers to dynamically place VNFs in an event-driven manner as end-user service requests are made. Secondly, it is essential to jointly consider routing paths during VNF placement. SFCs consist of multiple VNFs distributed across various nodes. A simple shortest-path algorithm may overlook critical factors such as the available computational capacity and delays at specific nodes, which could ultimately lead to higher E2E delays. Thus, a solution is required that not only dynamically places VNFs but also optimizes routing paths to minimize E2E delays.

An online VNF deployment strategy aimed at minimizing E2E delay was proposed in [16,20]. While these works address the placement of SFCs, they do not optimize the corresponding routing paths. In [12], joint VNF deployment and E2E delay minimization strategies were introduced. However, the current research requires tens of seconds for algorithm operation, making it challenging to handle user requests in real time. To fill the research gaps, we propose an online algorithm that jointly forms routing paths and deploys VNFs in an event-driven manner.

1.3. Contributions

The main contributions of this study are as follows:

- We design a system model for VNF deployment and routing path formation in SGIN. Based on this model, we derive a mixed integer non-linear programming (MINLP) problem to minimize the sum of E2E delays for all end users, considering that the service provider does not know the type or timing of the SFCs requested by end users.
- We propose an algorithm to form a minimum hop region (MHR) to reduce the computational complexity of finding a global optimal solution for the presented MINLP problem. This algorithm identifies candidate base stations suitable for routing path formation, differentiating them from non-candidate stations.
- We propose an online algorithm that deploys VNFs and forms routing paths immediately in response to end users' SFC calls. The algorithm first reduces propagation delay by forming routing paths and then minimizes the overall E2E delay by deploying VNFs on those paths. Our experimental results demonstrate that the proposed technique achieves 97% of the global optimal performance obtained through exhaustive search.

The remainder of this paper is structured as follows. Section 2 outlines the system model considered in this study. A detailed explanation of the proposed scheme is presented in Section 3, followed by the performance evaluation of the proposed algorithm in Section 4. Finally, the conclusions of this study are discussed in Section 5.

2. System Model

In this section, we describe the SGIN network model considered in this study. Subsequently, we explain the nodes and topology of the SGIN network. In addition, we model the characteristics of the SFCs and VNFs served through the SGIN network, along with the routing paths. Finally, we model the delay incurred when servicing SFCs from the service provider to the user through the routing paths formed.

2.1. Network Node Model

Figure 1 illustrates the SGIN network model considered in this study. The SGIN network contains four types of nodes: satellites, ground base stations (or routers), users, and service providers. In this paper, while the term "base station" is used, it can be interchangeably substituted with "router" in meaning. The set of service providers is denoted by $\mathcal{SP}$ and is assumed to be unique across the entire SGIN network. The set of satellites is represented by $\mathcal{S}$ and comprises n_s satellites. Similarly, the sets of ground base stations and users are denoted by $\mathcal{G}$ and $\mathcal{U}$, each comprising n_g and n_u nodes, respectively. The location of each node is described using two coordinate systems. One is the LLH coordinate system, which utilizes the latitude, longitude, and altitude, denoted by $(\lambda_n, \theta_n, h_n)$ for node n. The other is the ECEF coordinate system, centered at the Earth's center, denoted as (x_n, y_n, z_n) for node n. The distance between nodes n and m is defined as $r_{n,m}$. $r_{n,m}$ is calculated differently based on the node types n and m. If both n and m are ground-based nodes, the distance between them can be calculated using the Great-circle distance. Conversely, if one or both nodes are satellites, the distance between them is calculated using Euclidean distance. In summary, $r_{n,m}$ is defined as follows:

$$r_{n,m} = \begin{cases} E \cdot \tan^{-1}\left(\frac{\sqrt{\mathcal{H}(n,m)}}{\sqrt{1-\mathcal{H}(n,m)}}\right) & \text{if} \quad n, m \notin \mathcal{S} \\ \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2 + (z_n - z_m)^2} & \text{o.w} \end{cases} \tag{1}$$

where E denotes the radius of the Earth, and $\mathcal{H}(n,m)$ is defined as

$$\mathcal{H}(n,m) = \sin^2\left(\frac{\lambda_n - \lambda_m}{2}\right) + \cos\lambda_n \cdot \cos\lambda_m \cdot \sin^2\left(\frac{\theta_n - \theta_m}{2}\right) \tag{2}$$

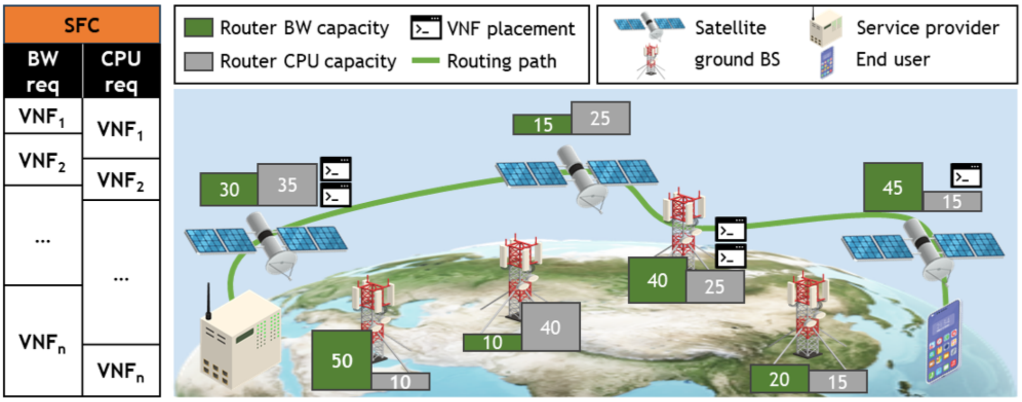


Figure 1. System model.

The communication link between nodes n and m is denoted by $l_{n,m}$. The set of all communication links is denoted by $\mathcal{L}$, and the number of elements in $\mathcal{L}$ is $(n_s + n_g + n_u + 1)^2$. We assume that a communication link is wired if both ends of the link are ground-based nodes and wireless if one or more ends are satellites. r_{th} is the maximum communication distance required to form a communication link. If the distance $r_{n,m}$ between two specific nodes n and m is less than or equal to r_{th}, a communication link is assumed to exist between the two nodes. Otherwise, if the distance between the two nodes exceeds r_{th}, it is assumed that there is no communication link between them and direct communication is not possible. Thus, the communication link $l_{n,m}$ between the two nodes can be defined as follows:

$$l_{n,m} = \begin{cases} 1, & \text{if} \quad r_{n,m} \leq r_{th}, \\ 0, & \text{o.w}, \end{cases} \qquad \forall n, m \in \mathcal{SP} \cup \mathcal{S} \cup \mathcal{G} \cup \mathcal{U}. \tag{3}$$

Finally, it is necessary to define the computing power of the base station nodes and the data rate of the communication links to model the computing and transmission delays. In SGIN, the role of a base station can be assumed by both ground base stations and satellites. The computing power of the base station node $n \in \mathcal{S} \cup \mathcal{G}$ is defined as P_n. In addition, the data rate of link $l_{n,m}$ is defined as $W_{n,m}$.

2.2. Service Function Chain Model

The set of SFCs provided by the service provider is denoted by $\mathcal{C}$, where n_c indicates the number of SFCs. Additionally, the set of VNFs is denoted by $\mathcal{F}$, where n_f is the number of VNFs. Moreover, $n_{n,f}$ represents the number of VNFs composing SFC n. Naturally, $n_{n,f}$ is a natural number less than or equal to n_f. Each VNF requires a certain amount of computing power for processing and generates a data size upon completion of the process. The computing power requirement of the VNF $f \in \mathcal{F}$ is defined as p_f, and the resulting data size after processing is defined as w_f.

The set of VNFs constituting SFC m is represented as $s_m = \left\{ f_{m,1}, f_{m,2}, \ldots, f_{m,n_{m,f}} \right\}$. The VNFs composing an SFC must be processed in a specific order. For convenience, SFC m is assumed to begin processing from $f_{m,1}$ and end processing with $f_{m,n_{m,f}}$. In this study, it is assumed that the VNFs composing an SFC are distinct from each other. However, the proposed algorithm can still operate even if the same VNF appears in different positions within the same SFC. When SFC m is invoked, the service provider must deploy all VNFs in s_m at the base station located between the service provider and user. It is assumed that VNFs are not installed on the service provider or user. Variable $\alpha_{n,f}$ is a binary variable that indicates whether VNF f is deployed at base station node n. If VNF f is deployed at node n, $\alpha_{n,f}$ is set to one; otherwise, it is set to zero. If a node exists in which all VNFs composing SFC m are deployed, the routing path for processing SFC m is designed to traverse nodes $\alpha_{n,f_{m,1}}$ and $\alpha_{n,f_{m,2}}$ in sequential order starting from the service provider and ending at the user. In summary, the set of rendezvous points for routing paths to process SFC m, denoted by $\mathcal{A}_m$, is defined as follows:

$$\mathcal{A}_m = \left\{ n \middle| n \in \mathcal{S} \cup \mathcal{G} \quad \text{where} \quad \alpha_{n,f} = 1 \quad \forall f \in s_m \right\}, \tag{4}$$

where $a_{m,i}$ indicates the i-th element of $\mathcal{A}_m$. Here, the set of rendezvous points $\mathcal{A}_m$ represents the nodes—either satellites ($\mathcal{S}$) or ground ($\mathcal{G}$) base stations—where the necessary VNFs are deployed to facilitate the SFC m. These nodes are crucial for effectively aggregating, routing, and processing data, ensuring that the service requirements for SFC m are met efficiently.

$\mathcal{A}_m$ represents the set of rendezvous points for servicing SFC m but does not represent the actual routing path. Therefore, we define set $\mathcal{B}$ to denote the actual routing path for servicing SFC m. Specifically, $\mathcal{B}_{m,i}$ is the set of communication links $l_{x,y}$ that represents the routing path from the node where $f_{m,i}$ is installed to the node where $f_{m,i+1}$ is installed. In other words, it represents the routing path from the rendezvous point $a_{m,i}$ to $a_{m,i+1}$ for servicing SFC m. If $f_{m,i}$ and $f_{m,i+1}$ are installed on the same node, that is, $a_{m,i} = a_{m,i+1}$, then $\mathcal{B}_{m,i}$ is an empty set. If $l_{x,y}$ is an element of $\mathcal{B}_{m,i}$, then $\beta_{x,y,m,i}$ is equal to one; otherwise, it is zero. $b_{m,i,j}$ is defined as the j-th element of $\mathcal{B}_{m,i}$.

2.3. Communication Delay Model

In this study, the range of E2E delay for servicing SFC m is defined as the time taken for the traffic flow of SFC m to travel from the service provider to the end user. E2E delays include propagation, computation, and transmission delays. Propagation delays are determined by the distance of $l_{x,y}$ and vary based on whether the communication link is wired or wireless. The propagation delay $d^p_{l_{x,y}}$ for the communication link $l_{x,y}$ is defined as

$$d^p_{l_{x,y}} = \begin{cases} \frac{r_{x,y}}{c} & \text{if} \quad l_{x,y} \quad \text{is wireless link} \\ \frac{r_{x,y} \cdot \rho}{c} & \text{if} \quad l_{x,y} \quad \text{is wired link} \end{cases} \tag{5}$$

c represents the speed of light and ρ denotes the refractive index of the wired link.

$d^c_n(f)$ is defined as the computing delay incurred when the base station node n processes the VNF f. The computing delay is determined by the computing power P_n of node n and the computing requirement p_f of VNF f. Specifically, the computing delay should increase inversely with the computing power of the node and directly with the computing requirements of the VNF. Additionally, in this study, it is assumed that if a base station installs multiple VNFs simultaneously, it allocates uniform computing resources to the installed VNFs. Considering these conditions, $d^c_n(f)$ can be defined as

$$d^c_n(f) = \frac{\sum_{f\prime \in \mathcal{F}} \alpha_{i,f\prime} \cdot p_f}{P_n} \tag{6}$$

$d^t_{l_{x,y}}(f)$ is defined as the transmission delay incurred when transmitting VNF f, whose processing has been completed through the communication link $l_{x,y}$. Similar to the computing delay, the transmission delay should increase inversely with the data rate of the link $W_{x,y}$ and directly with the data rate requirement of the VNF w_f. Additionally, if multiple VNFs need to be transmitted through the same communication link, the base station is assumed to distribute the bandwidth resources equally. Considering these factors, $d^t_{l_{x,y}}(f)$ can be defined as

$$d^t_{l_{x,y}}(f) = \frac{\sum_{m \in \mathcal{S}} \sum_{i \in s_m} \beta_{x,y,m,i} \cdot w_f}{W_{x,y}} \tag{7}$$

3. Proposed Algorithm

In this section, we define E2E delay using the delay functions defined earlier. In addition, we formulate a numerical problem of installing VNFs at base station nodes and forming routing paths to minimize the E2E delay of service provision through SFCs in SGIN. To solve this problem, we describe an algorithm for forming an MHR in Walker-delta environments. Finally, we introduce an algorithm for forming routing paths and determining the nodes for installing VNFs to service the SFCs in the formed MHR.

3.1. Problem Formulation

As mentioned previously, the scope of the E2E delay is defined as the traffic flow of the SFC from the service provider to the end user. The most natural form of E2E delay involves a sequential combination of computing, transmission, and propagation delays. For example, assuming servicing SFC m, first, transmission and propagation delays occur from the service provider to $a_{m,1}$. Second, a computing delay $d^c_{a_{m,1}}(f_{m,1})$ occurs when processing the VNF at node $a_{m,1}$. Following processing of VNF $f_{m,1}$, the transmission delay $d^t_{l_{x,y}}(f_{m,1})$ and propagation delay $d^p_{l_{x,y}}$ occur when transmitting traffic to $a_{m,2}$. This process is repeated until $a_{m,|s_m|}$ is used to compute the E2E delay. However, unlike computing delay, transmission and propagation delays have an additional factor to consider. Specifically, we must consider a scenario in which adjacent ordered VNFs within the SFC are installed on the same base station node. In this case, VNFs can be processed consecutively within a node without additional transmission or propagation processes, resulting in only a computing delay without transmission or propagation delays. Considering these conditions, the E2E delay function D_m incurred when the service provider services SFC m can be defined as follows:

$$D_m = \sum_{a_{m,i} \in \mathcal{A}_m} d^c_{a_{m,i}}(f_{m,i}) + \sum_{f \in s_m} \left(\min\left(1, |\mathcal{B}_{m,f}|\right) \right) \cdot \sum_{j \in \mathcal{B}_{m,f}} \left(d^t_{b_{m,f,j}}(f_{m,i}) + d^p_{b_{m,f,j}} \right) \tag{8}$$

The first summation represents the computing delay. The middle summation checks whether adjacent VNFs within the SFC should be processed on the same node. $|\mathcal{B}|$ indicates the number of elements in $\mathcal{B}$. If adjacent VNFs are installed at the same node, $|\mathcal{B}|$ becomes zero, indicating no transmission and propagation delay. Otherwise, the last summation enables the calculation of the transmission and propagation delays incurred by the actual routing path.

This study aims to minimize the E2E delay incurred when the service provider delivers all the SFCs by strategically deploying VNFs on base station nodes and forming routing paths for the SFCs. Accordingly, the following problem can be formulated.

$$\min_{\alpha,\beta} \quad \sum_{m \in \mathcal{C}} D_m \tag{9}$$

$$\text{subject to} \quad \sum_{a \in \mathcal{A}_m} \alpha_{m,f} = 1, \quad \forall f \in s_m, \tag{10}$$

$$\sum_{i \in s_m} \sum_{l_{x,y} \in \mathcal{B}_{m,i}} \beta_{x,y,m,i} = 1, \quad \forall l_{x,y} \in \mathcal{L}, \tag{11}$$

$$\alpha_{m,f} = \{0,1\}, \quad \forall f \in \mathcal{F}, m \in \mathcal{C}, \tag{12}$$

$$\beta_{x,y,m,f} = \{0,1\}, \quad \forall l_{x,y} \in \mathcal{L}, f \in \mathcal{F}, m \in \mathcal{C}. \tag{13}$$

The objective function (9) signifies that the E2E delay of all the SFCs provided by the service provider should be minimized. Constraint (10) implies that, within a single SFC, there should be at most one node where a VNF is installed. Importantly, this constraint does not mean that the same VNF cannot be installed on multiple nodes within the entire SGIN; rather, within the routing path for servicing a single SFC, only one node with that VNF should exist. The constraint (11) ensures that the traffic flow of an SFC does not redundantly pass through the same path, thereby avoiding redundant propagation delays. Constraints (12) and (13) are binary-variable constraint conditions for α and β, respectively. The objective function (9) represents a mixed-integer nonlinear programming problem that is NP-hard. Therefore, this paper proposes an online heuristic algorithm that aims to provide a solution close to the optimal, taking into account the NP-hard nature of the problem.

3.2. Minimum Hop Region Formulation

The first step in solving this problem is to select valid and invalid nodes to form the rendezvous points of the routing path to reduce the computational complexity of forming routing paths for servicing the SFCs. Valid rendezvous points refer to the nodes that do not cause redundant propagation delays between the source and the destination. This subsection introduces the concept of the MHR in Walker-delta and presents techniques for setting the area of the MHR to define the candidate nodes that can serve as rendezvous points.

Figure 2 illustrates the MHR for Walker-Delta. The MHR represents the area in the satellite network where a rational routing path can be formed between the source and destination. Specifically, *"rational"* means that physically redundant paths are not created when forming a routing path between the source and destination within the MHR. The conventional concept of MHR was presented for the Walker-star constellation [42]. The MHR was formed in a rectangular shape in a grid-based topology for the Walker-star. This is because, in Walker-star, there exists a SEAM area where adjacent orbits move in opposite directions, preventing the formation of ISLs. However, the inclination of the orbits became relatively parallel to the equator in the Walker-delta, eliminating the SEAM area. The technique for forming an MHR in Walker-star cannot be directly applied because a grid-based topology cannot be formed in the Walker-delta. Therefore, a new technique is required to define the MHR in the Walker-delta constellation.

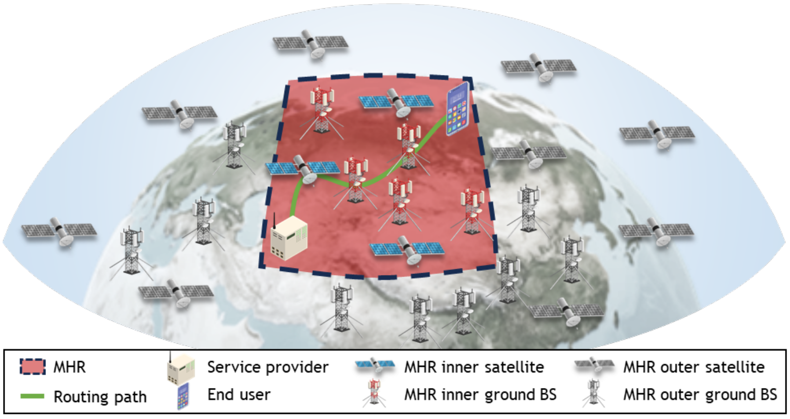


Figure 2. Minimum hop region for Walker-delta.

The MHR is determined based on the Great-circle path between the source and destination in the Walker-delta. Specifically, the MHR is determined based on the latitude and longitude of the endpoints of the Great-circle path. We checked whether both the source and destination are located in the eastern or western hemispheres longitudinally. It is assumed that negative values represent longitudes in the western hemisphere, whereas positive values represent longitudes in the eastern hemisphere. If both the source and destination are located in either the eastern or western hemispheres, then the longitudinal endpoints of the MHR are directly determined by the longitudes of the source and destination. For example, if the longitude of the source node is 30° east and the longitude of the destination node is 60° east, then the longitude of the MHR is $[30^\circ, 60^\circ]$. If one source and destination are in the eastern hemisphere and the other are in the western hemisphere, the absolute values of their longitudes are examined. If the sum of the absolute values of the longitudes of the source and destination is less than or equal to 180°, the Great-circle path between the two points passes through the longitude of 0°. In this case, the longitudinal endpoints of the MHR are set to the longitudes of the source and destination, as described earlier. Conversely, if the sum of the absolute values of the longitudes of the source and destination is greater than 180°, the Great-circle path between the two points passes through the longitude of -180°. In this case, the longitude of the MHR is defined to include the longitudes of the source and destination nodes up to -180°. By combining the four cases described above, the longitude range, θ_{MHR}, of the MHR can be defined as follows:

$$\theta_{MHR} = \begin{cases} [\min(\theta_s, \theta_d), \max(\theta_s, \theta_d)] & \text{if} \quad |\theta_s| + |\theta_d| \leq 180^\circ \quad \text{or} \quad \theta_s \cdot \theta_d \geq 0^\circ \\ [\min(\theta_s, \theta_d), -180^\circ) + [\max(\theta_s, \theta_d), -180^\circ] & \text{if} \quad |\theta_s| + |\theta_d| > 180^\circ \end{cases} \tag{14}$$

where θ_s represents the longitude of the source node and θ_d represents the longitude of the destination node.

In contrast to θ_{MHR}, which does not exceed the range between the longitudes of the source and destination, the latitude of the MHR λ_{MHR} may extend beyond the latitude range of the routing path endpoints. Owing to the characteristics of the Earth, moving the same longitude results in a shorter distance at higher latitudes than at lower latitudes. Therefore, λ_{MHR} is determined along the Great-circle path. Fortunately, it is not necessary to check the latitude of every point along the path because the Great-circle path is always either convex or concave. First, the latitude with the largest absolute value between the latitudes of the source and destination is determined. Note that we assume positive values for north latitude and negative values for south latitude. If both the source and destination are located in the northern hemisphere or the northern node is at a higher latitude, it is necessary to check how close the Great-circle path is to the north pole. Conversely, if both the source and destination are located in the southern hemisphere, or if the southern node is at a higher latitude, it is necessary to check how close the Great-circle path is to the south pole. The latitude range of the MHR is determined by iteratively specifying points along the Great-circle path, starting from the node at a higher latitude between the source and destination and checking the latitude and longitude. For convenience, we assumed that the source node is located at a higher latitude than the destination node. Additionally, we define the latitude and longitude of the source node as (λ_0, θ_0). The bearing angle between the source and destination nodes, ψ_0, is calculated before determining the latitude and longitude of the first exploration point. If we denote the latitude of the destination node as λ_d, the bearing angle to the destination node at the exploration point x, denoted as ψ_x, can be calculated as follows:

$$\psi_x = \arctan 2\left\{\sin\frac{\theta_d - \theta_x}{2} \cdot \cos\theta_d,\ \cos\lambda_x \cdot \sin\lambda_d - \sin\lambda_x \cdot \cos\lambda_d \cdot \cos\frac{\theta_d - \theta_x}{2}\right\} \tag{15}$$

After computing ψ_0 by using (15), we must calculate the latitude and longitude of the first exploration point, denoted as (λ_1, θ_1). If we have already obtained $(\lambda_{x-1}, \theta_{x-1}, \psi_{x-1})$ for the $(x-1)$th point, we can calculate the latitude of the x-th point as follows:

$$\lambda_x = \arcsin\left\{\sin\lambda_{x-1} \cdot \cos(r_{step}) \cdot \cos\lambda_{x-1} \cdot \sin(r_{step}) \cdot \cos\psi_{x-1}\right\} \tag{16}$$

where r_{step} represents the distance between the $(x-1)$-th and x-th exploration points; Naturally, setting r_{step} will expedite convergence but reduce the accuracy of λ_{MHR}, whereas setting r_{step} smaller will have the opposite effect. Finally, the longitude θ_x of the x-th point can be calculated as follows:

$$\theta_x = \arctan 2\left\{\sin\psi_{x-1} \cdot \sin(r_{step}) \cdot \cos\lambda_{x-1},\ \cos(r_{step}) - \sin\lambda_{x-1} \cdot \lambda_x\right\} + \theta_{x-1} \tag{17}$$

where λ_s represents the latitude of the source node. The process of determining the bearing angle, latitude, and longitude at each point is repeated iteratively by using Equations (15)–(17).

As described earlier, the iteration process can be stopped when λ_x begins to decrease below λ_{x-1} if the node with a higher latitude is located in the northern hemisphere because the Great-circle path is always either convex or concave. More specifically, if a node with a higher latitude is located in the northern hemisphere, the iteration can be halted when λ_x becomes smaller than λ_{x-1}, and λ_x can be set as the endpoint of λ_{MHR}. Conversely, if a node with a higher latitude is located in the southern hemisphere, the iteration can be stopped when λ_x becomes greater than λ_{x-1}. In addition, a polar region boundary exists for the Walker-delta. Therefore, if the absolute value of λ_x exceeds the polar region boundary, the endpoint of λ_{MHR} is set as the latitude of the polar region boundary, denoted by λ_{polar}.

Combining these explanations, the latitude range of the MHR, λ_{MHR}, can be defined as follows when the iteration ends at λ_{x+1}:

$$\lambda_{MHR} = \begin{cases} [\min(\lambda_s, \lambda_d), \lambda_x] & \text{if} \quad \max(\lambda_s, \lambda_d) + \min(\lambda_s, \lambda_d) \geq 0^\circ \\ [\lambda_x, \max(\lambda_s, \lambda_d)] & \text{if} \quad \max(\lambda_s, \lambda_d) + \min(\lambda_s, \lambda_d) < 0^\circ \\ \left[\min(\lambda_s, \lambda_d), \lambda_{polar}\right] & \text{if} \quad \lambda_{x+1} \geq \lambda_{polar} \\ \left[\lambda_{polar}, \max(\lambda_s, \lambda_d)\right] & \text{if} \quad \lambda_{x+1} < \lambda_{polar} \end{cases} \tag{18}$$

We can determine the ranges of latitudes and longitudes for the MHR, denoted by $(\lambda_{MHR}, \theta_{MHR})$. By defining the MHR area, we can avoid generating redundant propagation delays in the Walker-delta and distinguish candidate base stations that can be considered as rendezvous points. Algorithm 1 presents the pseudocode that illustrates the process of determining $(\lambda_{MHR}, \theta_{MHR})$. Specifically, each condition in (18) directly matches the corresponding steps in Algorithm 1. The first condition in (18) is handled in line 28 of Algorithm 1. The second condition in (18) is addressed in line 25. The third condition corresponds to line 20. Finally, the fourth condition is reflected in line 18.

Algorithm 1: Minimum Hop Region Determination

```
   Input :Latitude and longitude for source and destination, $(\lambda_s, \theta_s), (\lambda_d, \theta_d)$
 1 if $|\theta_s| + |\theta_d| > 180^\circ$ then
 2     $\theta_{MHR} = [\min(\theta_s, \theta_d), -180^\circ) + [\max(\theta_s, \theta_d), -180^\circ]$
 3 else
 4     $\theta_{MHR} = [\min(\theta_s, \theta_d), \max(\theta_s, \theta_d)]$
 5 end
 6 if $|\lambda_s| \leq |\lambda_d|$ then
 7     $(\lambda_0, \theta_0) = (\lambda_s, \theta_s)$
 8 else
 9     $(\lambda_0, \theta_0) = (\lambda_d, \theta_d)$
10 end
11 $x = 0$
12 while True do
13     $\psi_x$ can be calculated using (15)
14     $\lambda_x$ can be calculated using (16)
15     $\theta_x$ can be calculated using (17)
16     if $|\lambda_x| > |\lambda_{polar}|$ then
17         if $\lambda_0 \leq 0^\circ$ then
18             $\lambda_{MHR} = \left[\lambda_{polar}, \max(\lambda_s, \lambda_d)\right]$
19         else
20             $\lambda_{MHR} = \left[\min(\lambda_s, \lambda_d), \lambda_{polar}\right]$
21         end
22         Break Loop
23     else
24         if $\lambda_0 \leq 0^\circ$ and $\lambda_x > \lambda_{x-1}$ then
25             $\lambda_{MHR} = [\lambda_{x-1}, \max(\lambda_s, \lambda_d)]$
26             Break Loop
27         else
28             $\lambda_{MHR} = [\min(\lambda_s, \lambda_d), \lambda_{x-1}]$
29             Break Loop
30         end
31     end
32     $x = x + 1$
33 end
```

3.3. VNF Placement and Routing Path Formulation

The MHR formation process described in the previous subsection can be performed by setting the source to the location and destination of the service provider at the end user's location. LEO satellites and ground base stations within the formed MHR area do not introduce redundant propagation delay, making them rational router candidates. Once candidate base stations are selected, the service provider must minimize E2E delay, form routing paths, and deploy VNFs to nodes along the path to process SFCs for end users. This subsection explains the process of forming routing paths and deploying VNFs for the service provider to process the SFCs by minimizing the E2E delay.

If the service provider decides to install SFCs that have not had their VNFs installed at any base station, it must formulate the physical routing paths before installing them. When forming routing paths, the service provider should consider the following fact: *"Transmission and computing delay can be adjusted based on how VNF installation is performed, but propagation delay remains unchanged once the physical routing path is determined"*. Therefore, the service provider formulates a routing path to minimize E2E propagation delay. Specifically, the service provider sets the propagation delay of the links between base stations, denoted by $l_{x,y} \cdot d^{p}_{l_{x,y}}$, as the weights of the communication links. The service provider then forms several routing paths using Yen's algorithm to minimize the sum of the weights by excluding communication links with zero weights. If multiple routing paths that can minimize the propagation delay are found using Yen's algorithm, the VNF placement algorithm, described later, is executed for each path. Naturally, there exists a tradeoff in that finding many routing paths results in a performance that approaches the global optimum at the expense of increased computational complexity.

Once the candidate nodes for VNF installation have been determined, it is necessary to decide how to deploy the VNFs. For instance, all the VNFs could be installed at the first base station on the routing path (the base station closest to the service provider). Alternatively, the VNFs could be evenly distributed among the nodes along the routing path. Given the routing path, computing and bandwidth resources of the nodes, and computing and data rate requirements of the VNFs, the simplest method to minimize the E2E delay is to explore all possible VNF deployment scenarios and perform an exhaustive search for the E2E delay. However, this exhaustive search process is computationally and temporally intensive, making it difficult to quickly determine the optimal VNF deployment scenario that minimizes the E2E delay. A VNF placement algorithm that can approximate the global optimum without exhaustively searching all scenarios is required to expedite VNF deployment.

The service provider must understand the impact on the E2E delay when VNFs are deployed relatively forward (closer to the service provider) and backward (closer to the end user) within the routing path to deploy VNFs without an exhaustive search. The service provider defines the following rule for VNF deployment to prevent redundant propagation delay: *"If the y-th VNF is installed at the x-th base station, subsequent VNFs can only be installed at nodes located later than the x-th base station"*. For example, if the second VNF is installed at the third base station along the routing path, subsequent VNFs, such as the third and fourth, can only be installed at the third, fourth, fifth, and subsequent base stations. If the second VNF is installed at the third base station and the third VNF is installed at the second base station, redundant propagation delays would occur because the link between the third and second base stations is used multiple times.

The service provider then determines the sequential installation from the first VNF. Assuming that the first VNF is installed at the x-th base station, the total sum of the transmission delays from the service provider to the x-th base station must be calculated. Specifically, the total sum of transmission delay is $\frac{w_0}{W_{l_{0,1}}} + \frac{w_0}{W_{l_{1,2}}} + \ldots + \frac{w_0}{W_{l_{x-1,x}}}$ considering the communication link between the x-th and x'-th base stations along the routing path as $l_{x,x'}$. Here, w_0 represents the data rate requirement before the first VNF is processed, and $l_{0,1}$ denotes the communication link between the service provider and the first base station. Generalizing this by assuming that the $(y-1)$-th VNF f_{y-1} is installed at the x'-th base station, the transmission delay $\hat{d}^t_{x',x}(f_{y-1}, f_y)$ required to install the y-th VNF f_y, at the x-th base station is as follows:

$$\hat{d}^t_{x',x}(f_{y-1}, f_y) = \sum_{i=x'}^{x-1} \left(\frac{w_{f_{y-1}}}{W_{l_{i,i+1}}} \right) \tag{19}$$

Equation (19) naturally increases when the VNFs are placed further along the routing path. In other words, if the service provider wants to minimize (19) to install the VNFs, there is a tendency to be installed toward the front of the routing path. If the data rate requirements of the VNFs after the y-th VNF are low, it may be appropriate to place VNF f_y toward the front. However, if the data rate requirements of the VNFs after the y-th VNF are higher than w_{f_y}, it may be advantageous to place VNF f_y toward the back of the routing path and overlap multiple VNF installations at the base stations placed at the back of the path. In summary, the impact of the transmission delay caused by VNFs after f_y must be considered when installing f_y at the x-th base station. Unfortunately, the exact locations where the remaining VNFs will be installed cannot be known, unlike in (19), making it impossible to accurately predict the transmission delay. Moreover, it is unsuitable for calculating all possible cases in which VNFs after f_y can be installed at base stations beyond the x-th base station.

Algorithm 2 presents the pseudocode for estimating the transmission delay caused by subsequent VNFs when installing VNF f_y at the x-th base station.

Algorithm 2: Estimation for Potential Transmission Delay

```
Input: Set of VNFs of SFC after $f_y$, Set of base station after $x$
Set search starting point as $x$
Set $\hat{n}_f$ as the number of remaining VNFs
Set the search ending point as the base station $\hat{n}_f$-th from the end user.
foreach $f$ of Set of VNFs do
    Set $\hat{n}_b$ as the number of base stations from the starting point to the end user
    if $\hat{n}_b \leq \hat{n}_f$ then
        Assume $f$ is installed at base station $x$
        Reset the search starting point as $x + 1$-th base station.
    else
        Find the $x'$-th base station with the maximum $\frac{w_f}{W_{l_{x,x+1}}}$
        Assume $f$ is installed at base station $x'$
        Reset the search starting point as $x'$-th base station
    end
    Reset $\hat{n}_f$ as the number of remaining VNFs
    Reset the search ending point as the base station $\hat{n}_f$-th away from the end user
end
We estimated $\tilde{d}^t_x(f_v)$ as the potential transmission delay
```

The potential transmission delay is estimated greedily. In addition, the algorithm predicts the worst-case transmission delay that can occur when VNF f_y is installed at the x-th base station. The aim is to encourage the placement of VNFs on lower-latency base stations by predicting the potential worst-case transmission delays. All the VNFs must be installed at different base stations to calculate the worst-case transmission delay. If the number of remaining base stations is less than the remaining number of VNFs, the remaining VNFs are sequentially assumed to be installed, starting at the base station of the search. Otherwise, the algorithm assumes the installation of the VNF at the base station, which causes the maximum transmission delay. Once these VNF placement scenarios are established, the algorithm estimates the maximum potential transmission delay $\tilde{d}^t_x(f_v)$. $\tilde{d}^t_x(f_v)$ obtained using Algorithm 2 tends to decrease as the VNFs are placed further back along the routing path.

Equation (19) encourages the service provider to consider installing the y-th VNF in the front of the routing path. Conversely, Algorithm 2 encourages the placement of the y-th VNF toward the rear of the routing path. Therefore, the service provider computes $\hat{d}^t_{x',x}(f_{y-1}, f_y)$ and $\tilde{d}^t_x(f_v)$ for each possible base station x when deciding where to install the y-th VNF. The sum of these two transmission delays is considered as the effect of installing the y-th VNF at the x-th base station.

The impact of the computing delay when installing the y-th VNF at the x-th base station must be considered, similar to the transmission delay. The most naive approach is to install a VNF on the base station with the highest computing power P_x. However, two points must be considered. The first point to consider is that if a base station with high computing power is located at the rear of the routing path, the remaining VNFs may be excessively overlapped with the VNFs. When a base station installs multiple VNFs simultaneously, its computing power is evenly distributed among the VNFs. Therefore, excessive overlapping of VNFs may amplify the computing delay. Thus, the computing delay when installing the y-th VNF at the x-th base station is calculated by assuming that the remaining VNFs are evenly distributed among the remaining routers and then amplifying the computing requirements p_f of the VNF f_y. Similarly, the second point to consider is that if the y-th VNF is additionally installed at the base station x, which already has n' VNFs installed, the computing delay should be calculated as $\frac{(n'+1)\cdot p_y}{P_x}$. The service provider calculates the impact of the computing delay when installing the y-th VNF at the x-th base station using the aforementioned method.

The service provider calculates the sum of the computing and transmission delays incurred when installing VNF f_y at each base station along the routing path. The service provider then installs VNF f_y at the base station with the lowest total delay sum. This process is repeated for the $(y+1)$-th VNF. After installing the final VNF, adjustments are made to the values of $d^p_{l_{x,x'}}$ for the subsequent SFC installation. Specifically, the computing and transmission delays that are unavoidably caused by the installed SFC are considered as propagation delays that guide the subsequent SFC to avoid the corresponding base station and routing path. The transmission delay $d^t_{l_{x,x+1}}(f_y)$ incurred on communication link $l_{x,x+1}$ owing to the installation of VNF f_y is added directly to $d^p_{l_{x,x+1}}$. The computing delay $d^c_x(f_y)$ incurred at base station x owing to the installation of VNF f_y is added to all $d^p_{l_{x,x'}}$ associated with base station x. Subsequently, the service provider considers the updated propagation delay when forming a new routing path to serve a new SFC. Algorithm 3 represents the pseudocode for the aforementioned routing path formation and VNF placement algorithm.

Algorithm 3: Joint Routing Path Formulation and VNF Placement

```
Input: Set of VNFs of SFC, Set of the base station of routing path
Set $i = 0$
foreach $f$ of Set of VNFs do
    foreach Set of base station from i-th do
        Estimate transmission and computing delay
        Add the two delays
    end
    $f$ is installed at the base station with the minimum sum of two delays
    Reset $i$ as the index of the base station, where $f$ is installed
end
Update the propagation delay based on the VNF placement scenario
```

4. Simulation Results

In this section, we evaluate the performance of the joint routing path and VNF placement algorithm proposed in this paper. Here, we describe the simulator environment used in our experiments. We evaluated the performance by varying these three parameters.

The first parameter is the Great-circle distance between the service provider and the end user. The second parameter involves changing the number of SFCs provided by the service provider and the number of VNFs composing the SFC. The third parameter involves changing the number of LEO satellites and ground base stations composing the SGIN.

4.1. Simulation Environment

We implemented three benchmark schemes to compare the performance of the proposed scheme. The specific descriptions of each scheme are as follows:

- *Exhaustive search* (ES): In this scheme, the service provider considers all possible routing paths within MHR and VNF placement cases in the SGIN to find the one that minimizes the E2E delay. This scheme serves as a reference to evaluate how closely the proposed scheme approaches the global optimum, despite its high computational and time complexity.
- *Proposed scheme* (PR): In this scheme, the service provider identifies candidate base stations using Algorithm 1 and places VNFs using Algorithm 3 within the three shortest routing paths by using Yen's algorithm.
- *Greedy approach* (GR): In this scheme, the service provider minimizes the computational complexity by using the Dijkstra algorithm to form routing paths with minimal propagation delay. Subsequently, VNFs are placed on the base station with the lowest combined computing and transmission delay along the formed routing path. For example, if the delay incurred for processing the first VNF is minimal at the second base station along the path, the first VNF is installed at the second base station.

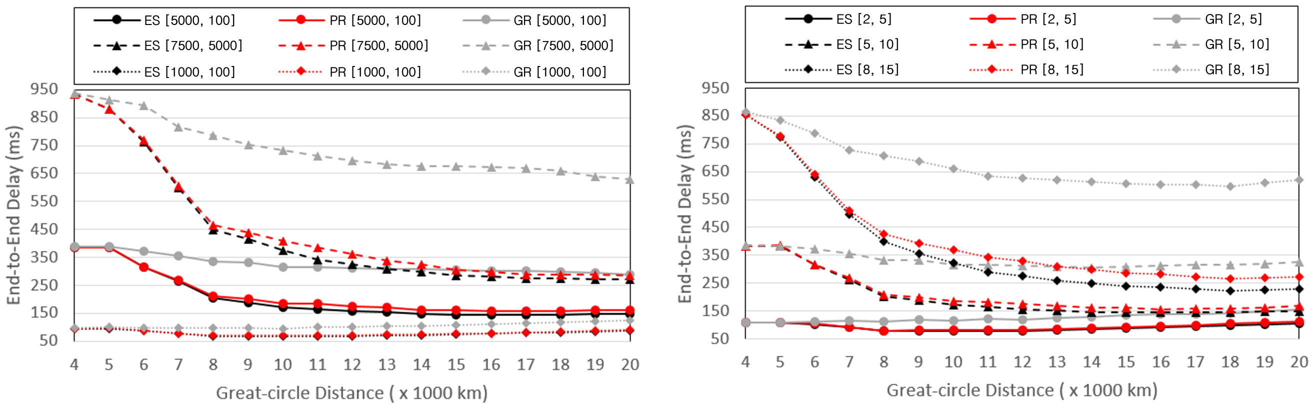
Table 1 lists the system parameters used in the simulations. The system parameters are configured based on [33,34,43]. It is assumed that the service provider is located in Seoul, and the end user is randomly positioned at a specified Great-circle distance from Seoul. All the performance values displayed in the graphs were determined as the average of 10,000 experiments, varying the resource requirements of the VNFs, the locations of the base stations constituting the SGIN, and the computing and communication resources of the base stations.

Table 1. Simulation parameters.

Notation	Description	Value
n_s	The number of satellites	5
n_u	The number of users	10
n_g	The number of ground base stations	80
n_f	The number of VNFs	20
n_c	The number of SFCs	$[2, 8]$
p_f	The computing power requirement of the VNF f	$U[100, 7500]$
P_n	The computing power of base station node n	$U[250, 500]$ (for satellite) $U[500, 1000]$ (for ground)
$W_{x,y}$	The data rate of the link	$U[500, 1000]$
w_f	The resulting data size after processing f	$U[100, 7500]$
r_{th}	The maximum communication distance	4000 (km)

4.2. Impacts of Great-Circle Distance

We compared the performance of the three schemes by varying the Great-circle distance between the service provider and the end user. The Great-circle distance is within the range of r_{th} to $20,000$ km with intervals of 1000 km. Figure 3 presents graphs depicting the variation in the E2E delay with respect to the Great-circle distance. Figure 3a measures the E2E delay while changing the computing capacity of the VNFs, p_f, and the size of the data after processing, w_f. The values of p_f and w_f for the VNFs are randomly determined within predefined ranges using a uniform distribution. Figure 3b measures the E2E delay while varying the number of SFCs and the number of VNFs composing each SFC. We represent the scenarios with x SFCs and y VNFs per SFC as $[x, y]$.



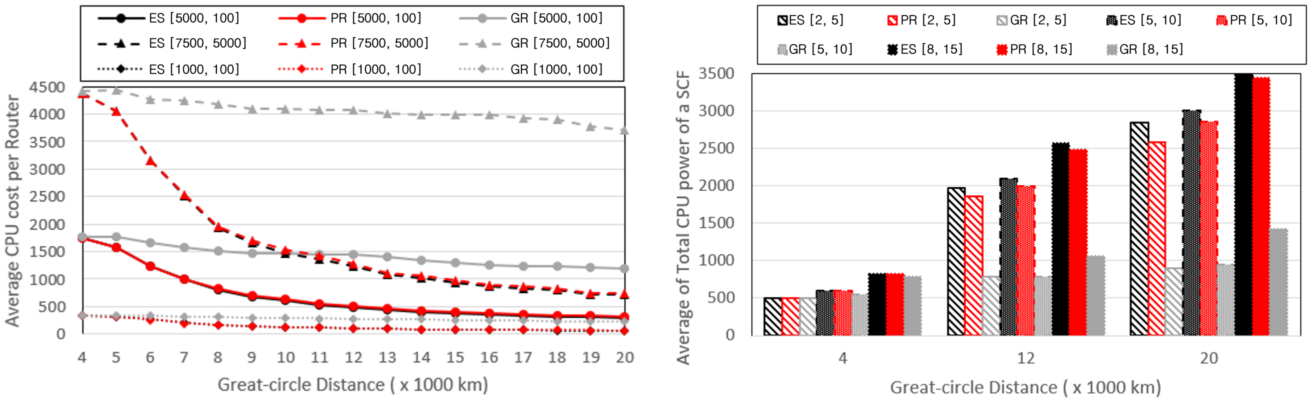
(**a**) (**b**)

Figure 3. E2E delay based on the Great-circle distance and (**a**) p_f and w_f, (**b**) n_c and n_f.

In Figure 3a, it can be observed that, regardless of the scheme type, the E2E delay increases as the distribution of computing and communication resources required to process the VNFs increases. This is because the computing and transmission delays incurred in processing and transmitting the VNFs increase proportionally with the resource requirements. Another notable finding is that despite the increase in the Great-circle distance, a general trend of decreasing E2E delay exists. This can be attributed to the increase in the number of base stations included in the MHR as the Great-circle distance increases. The number of base stations included in the routing path increases with the number of base stations included in the MHR. The VNFs can be distributed across multiple base stations as the number of base stations that form a routing path increases. Consequently, the computing delay decreases as the Great-circle distance increases, resulting in a reduction in the E2E delay. However, a different phenomenon occurs when the Great-circle distance is greater than 10,000 km in the $[1000, 100]$ scenario. This is because the computing power of the base stations in the routing path is already sufficient, causing an increase in the propagation delay to outweigh the decrease in the computing delay. Consequently, the E2E delay increases slightly. The higher E2E delay observed in PR compared to ES can be attributed to the discrepancy between the estimated transmission delay calculated using Algorithm 2 and the actual transmission delay. Despite this discrepancy, the similarity in E2E delay between PR and ES indicates that Algorithm 2 effectively predicts the potential transmission delay. Moreover, the PR can quickly approximate values close to the optimum with a complexity of $O(n_{m,f})$, whereas the ES exhaustively searches for all possibilities. This demonstrates that while PR may have a slightly higher E2E delay, its efficiency in approximating the optimum solution makes it a practical alternative to ES. For PR, when $[1000, 100]$, it shows 97% similarities to the E2E delay of ES. In this case, it implies that the computing requirement of VNF is low, and the resulting data size after processing is also small. Thus, the proportion of computing and transmission delay becomes minimal, making the impact of propagation delay more significant. As a result, placing the VNF on the shortest physical path becomes close to the optimal solution in most cases, leading to the observed 97% similarity. In other general experimental results, we observed that PR, on average, shows 90% similarity to the E2E delay of ES. The GR scheme exhibits the highest E2E delay when the base station suitable for installing VNFs is located at the rear of the routing path, resulting in the duplication of all VNFs in the rear. In Figure 3b, it is also observed that as the number of SFCs and VNFs increases, the computing and transmission delay increases, increasing the E2E delay. In both graphs, the difference in

E2E delay between the schemes is not significant when the Great-circle distance is small. This is because fewer base stations exist within the MHR, resulting in fewer routing paths and fewer VNF installation cases.

Figure 4 depicts graphs showing the variation in the CPU cost with respect to the Great-circle distance. The CPU cost of a single base station is defined as the product of the number of installed VNFs and the sum of p_f values of the VNFs. Figure 4a illustrates the graph that shows the average CPU costs assigned to a single base station. Figure 4b illustrates the graph representing the averages of the total CPU power of the base stations serving SFCs. Importantly, base stations without installed VNFs are not included in the averages. Figure 4a exhibits a trend similar to that shown in Figure 3a. This is because the CPU cost assigned to the base stations has an immediate impact on computing delay. As the MHR increases proportionally with the Great-circle distance, the number of base stations that form the routing path also increases, leading to the distribution of VNFs across multiple base stations and subsequently reducing the CPU cost of the base stations. Similarly, the total CPU power of the base stations that serve the SFCs increases proportionally with the Great-circle distance, as shown in Figure 4b. Additionally, Figure 4b shows that the total CPU power increases proportionally with the number of SFCs and VNFs. This is because, as the number of VNFs increases, the number of base stations installing VNFs to process the SFCs also increases. The PR scheme exhibits a slightly higher CPU cost and slightly lower CPU power than the ES, which examines all routing paths. In the case of the GR scheme, the occurrence of duplicating multiple VNFs in the rear of the routing path increases the CPU cost and decreases the total CPU power.

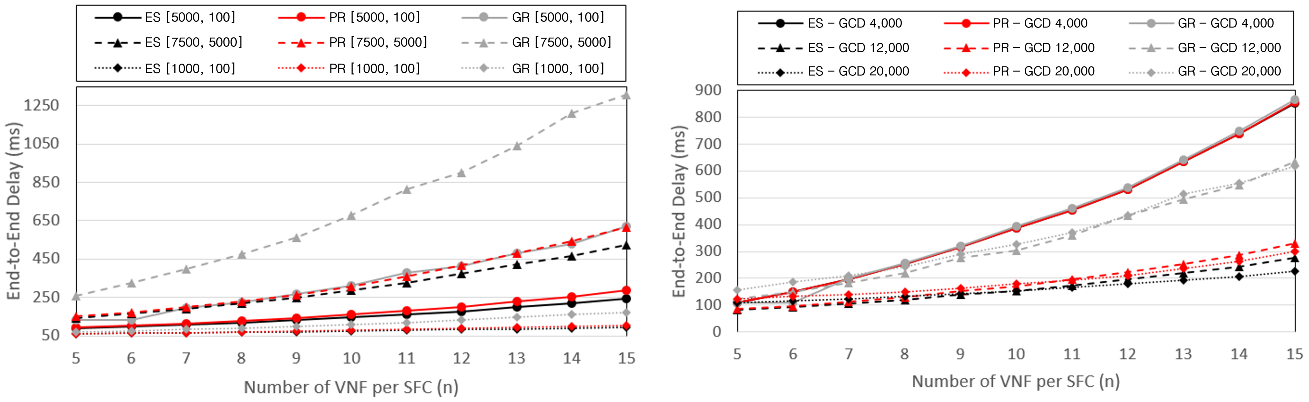


(**a**) (**b**)

Figure 4. Performance changes of (**a**) average CPU cost per router and (**b**) total CPU power to process an SFC based on the Great-circle distance.

4.3. Impacts of Configuration of SFCs

We compare the performance by varying the number of SFCs provided by the service provider and the number of VNFs composing each SFC; the number of VNFs varies from five to fifteen, and the number of SFCs varies from two to eight. Figure 5 depicts graphs of the E2E delay with changes in the number of VNFs composing the SFCs. Figure 5a illustrates the change in E2E delay while varying the range of resource requirements for the VNFs, with five SFCs and a Great-circle distance of 15,000 km. Figure 5b represents the change in the E2E delay while varying the Great-circle distance between the service provider and end user, with the range of resource requirements for VNFs set to $U[5000, 100]$. Similar to the reasons in Figures 3a and 4a, as the resource requirements of the VNFs increase, the computing and transmission delays also increase, increasing the E2E delay. Additionally, an increase in the number of VNFs results in a proportional increase in the E2E delay. Unlike the GR scheme, the PR scheme exhibits a relatively gradual increase in E2E delay with an increase in the number of VNFs because it prevents the duplication of VNF installations in the rear part of the routing path using Algorithm 2. Therefore, it exhibits E2E delays that are relatively similar to those of the ES. All three schemes exhibit similar performances in the case of a Great-circle distance of 4000 km, where the number of base stations in the MHR area is small. However, the PR scheme drastically reduces the E2E delay, similar to the ES scheme, as the MHR area increases, whereas the GR scheme shows a relatively marginal decrease in the E2E delay owing to the clustering of VNFs in the rear part of the routing path.

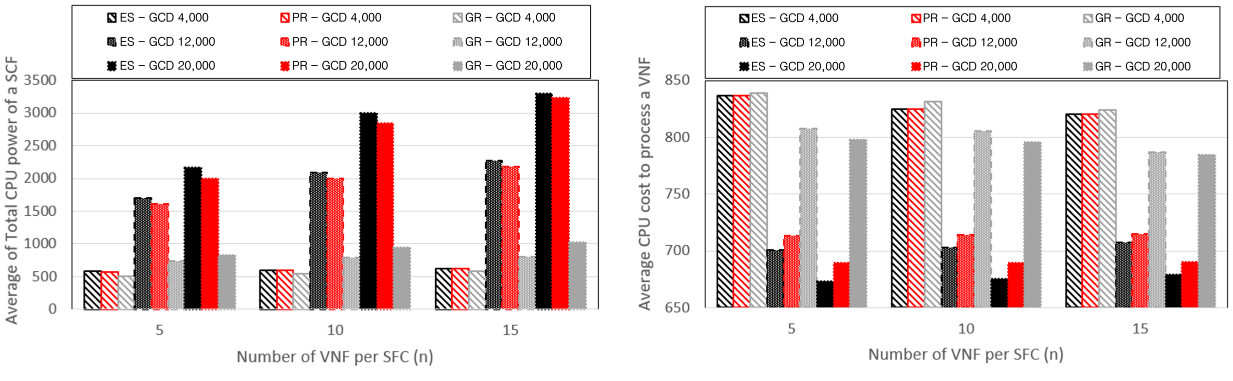


(**a**) (**b**)

Figure 5. E2E delay based on the number of VNFs per SFC and (**a**) p_f and w_f; (**b**) Great-circle distance.

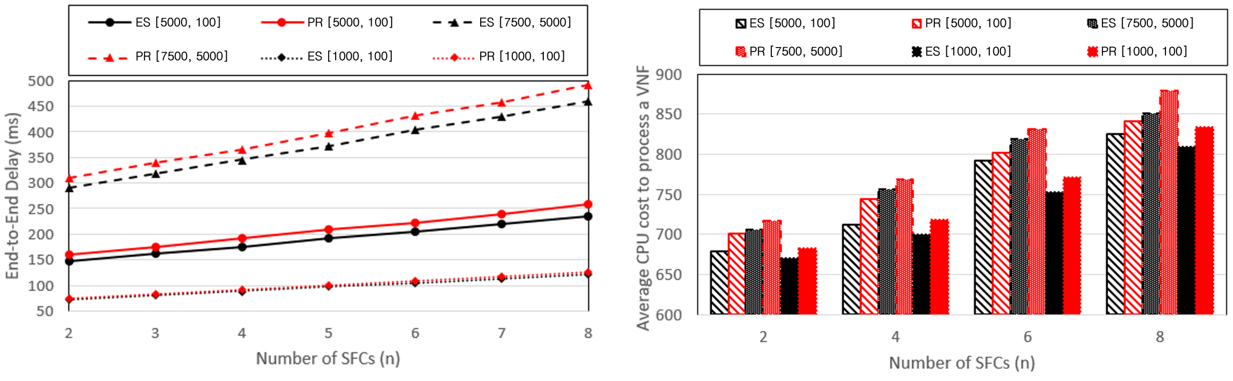
Figure 6 depicts graphs showing the performance related to the computing power while varying the Great-circle distance. Figure 6a represents the average total CPU power of the base stations deployed to process the VNFs composing the SFCs. Base stations without installed VNFs are not included in the computations. Figure 6b illustrates the average CPU cost incurred to process a single VNF, where the CPU cost for processing VNFs is defined as the quotient of the CPU cost per base station and the number of VNFs installed. The total CPU power required to service the SFCs increases with the number of VNFs and the length of the Great-circle distance. With a larger number of VNFs, the schemes tend to be distributed across multiple base stations, resulting in an increase in CPU power. Similarly, an increase in the Great-circle distance induces the dispersion of VNFs across multiple base stations, increasing the CPU power. PR prevents the excessive duplication of VNF installations on the same base station through Algorithm 3, thereby ensuring higher CPU power compared with GR. As the Great-circle distance increases, the VNFs are distributed across multiple base stations, decreasing the CPU cost required to process a single VNF. However, it can be observed that the CPU cost incurred for VNF processing is not significantly affected by the number of VNFs composing the SFCs. This is because the PR scheme determines the distribution of VNFs across multiple base stations based on the CPU cost using Algorithm 3. The PR scheme tends to install additional VNFs on base stations with lower CPU costs while avoiding additional installations on base stations with higher CPU costs, resulting in similar CPU costs regardless of the number of VNFs.

Figure 7 illustrates graphs that compare the performance while varying the number of SFCs provided by the service provider. The number of VNFs composing the SFC was fixed at ten. Figure 7a presents graphs that compare the E2E delay with changes in the distribution of the VNF resource requirements. Figure 7b represents a graph that depicts the CPU cost incurred to process a single VNF. There exists a tendency for both the E2E delay and CPU costs to increase proportionally with an increase in the number of SFCs. This is attributed to the increase in the total number of VNFs installed on the SGIN as the number of SFCs increases, increasing the computing and transmission delays.



(**a**) (**b**)

Figure 6. Performance changes of (**a**) the total CPU power of an SFC, and (**b**) the average CPU cost to process a VNF according to the number of VNFs per SFC.



(**a**) (**b**)

Figure 7. Performance changes of (**a**) E2E delay and the (**b**) average CPU cost based on the number of VNFs per SFC.

4.4. Impacts of the Number of Base Station Nodes

We compare the performances of the three schemes by varying the number of satellites and ground base stations in the SGIN. For convenience, we denote the number of satellites n_s and ground base stations n_g in the graphs as $[n_s, n_g]$. Figure 8 presents graphs that compare the E2E delay while varying the SGIN configuration and the number of VNFs that compose the SFC. The number of SFCs provided is fixed at eight. Figure 8a and Figure 8b represent graphs for cases where the resource requirements of the VNFs are $U[5000, 100]$ and $U[7500, 5000]$, respectively. As the number of base stations increases, the number of base stations within the MHR also proportionally increases. Consequently, the number of base stations that compose the routing path also increases proportionally, leading to a reduction in the E2E delay owing to the decreased computing delay. In addition, there exists a higher probability of forming routing paths similar to the shortest physical distance as the number of base stations increases, leading to a decrease in propagation delay.

Figure 9 shows graphs that depict the CPU cost required to process a single VNF while varying the SGIN configuration. Figure 9a and Figure 9b represent graphs for cases where the resource requirements of the VNFs are $U[5000, 100]$ and $U[7500, 5000]$, respectively. Unlike the number of VNFs, it can be observed that as the number of base stations increases, the CPU cost required to process a single VNF decreases. This is because the number of base stations that form a routing path is affected by the SGIN configuration. When more base stations are included in the routing path, additional opportunities to distribute VNFs exist, decreasing the CPU cost required to process a single VNF. Furthermore, even as the number of base stations increases, the performance difference between the PR and ES schemes remains relatively consistent. However, it can be concluded that the PR scheme becomes more advantageous as the number of base stations in the SGIN increases, considering that the complexity of the ES scheme increases more rapidly than that of the PR scheme as the number of base stations increases.

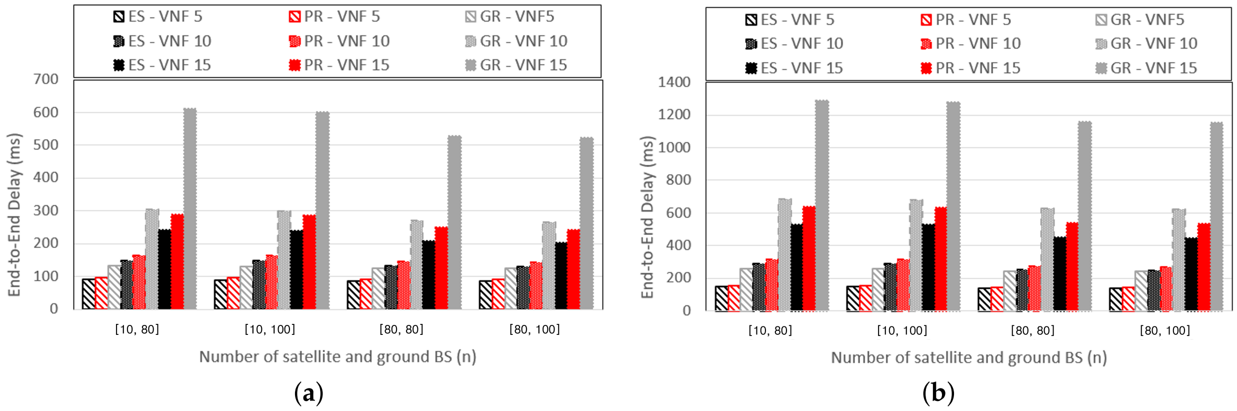


Figure 8. E2E delay according to the SGIN configuration with (**a**) $p_f = U[5000, 100]$, (**b**) $p_f = U[7500, 5000]$.

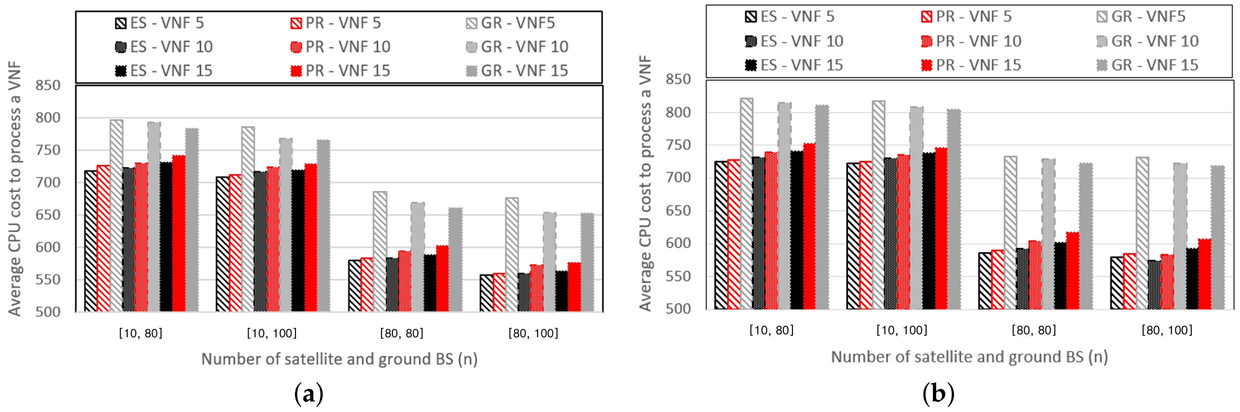


Figure 9. Average CPU cost to process a VNF according to the SGIN configuration with (**a**) $p_f = U[100, 5000]$, (**b**) $p_f = U[5000, 7500]$.

4.5. Comparison with Other Methods

At present, there exists one research study that jointly performs VNF placement and routing path formation to minimize E2E delay [12]. The authors aim to minimize E2E service latency by jointly optimizing VNF placement and routing. To achieve this, they proposed a time-expansion-based decoupled greedy (TEDG) algorithm, which is a heuristic algorithm that aims to reduce computational complexity while achieving near-optimal performance. For the performance comparison, they introduced the following four methods:

- *TEDG with the modified max-min normalization* (MM): The MM method aims to better utilize available resources by adjusting the weights to improve the selection of routing paths and the allocation of VNFs, focusing on optimizing both latency and resource usage.
- *TEDG with equal weight* (EW): A variation of the TEDG algorithm where the cost of all available links, including communication and stay links, is assigned an equal weight of 1.
- *Decoupled greedy algorithm* (DG): A modified version of a decoupled greedy algorithm, which allocates resources to services based on a configuration period rather than a time slot.

- *Genetic algorithm* (GA): A genetic algorithm applied to solve the joint VNF placement and routing planning problem. This approach is also used in similar contexts for optimizing resource allocation in cloud centers.

We intend to compare the proposed MM, EW, DG, and GA algorithms from the referenced paper alongside the ES and PR algorithms from our experiments to demonstrate the superior performance of our approach. The experimental environment and parameters were the same as those used in the work.

Figure 10 is a performance analysis graph of end-to-end delay as a function of computing power. As shown, the ES algorithm consistently achieves the lowest E2E delay since it finds the optimal solution. In the case of the PR algorithm, it begins to outperform other methods starting from 300 units of computing power. The proposed PR method places VNFs based on the predicted transmission delay, which is calculated using computing power divided by transmission data rates. When computing power is low, the prediction accuracy of our method decreases, resulting in significant performance degradation compared to ES. However, as computing power increases, the accuracy improves, allowing the PR method to achieve performance similar to that of the ES method.

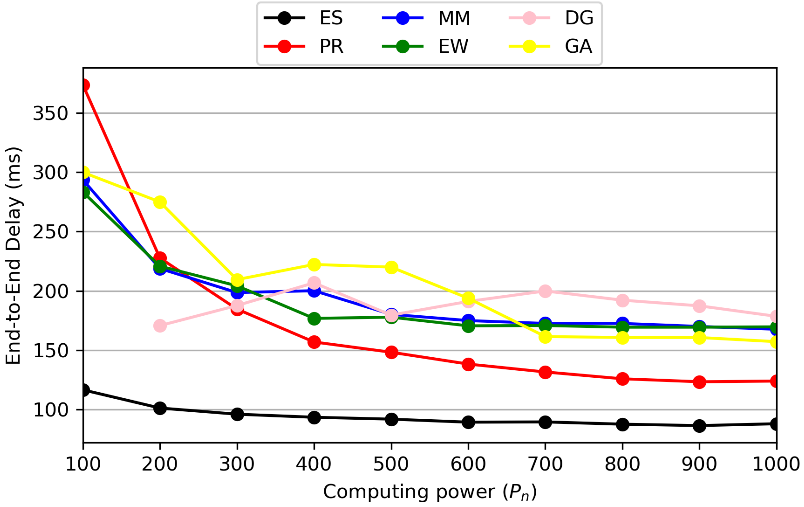


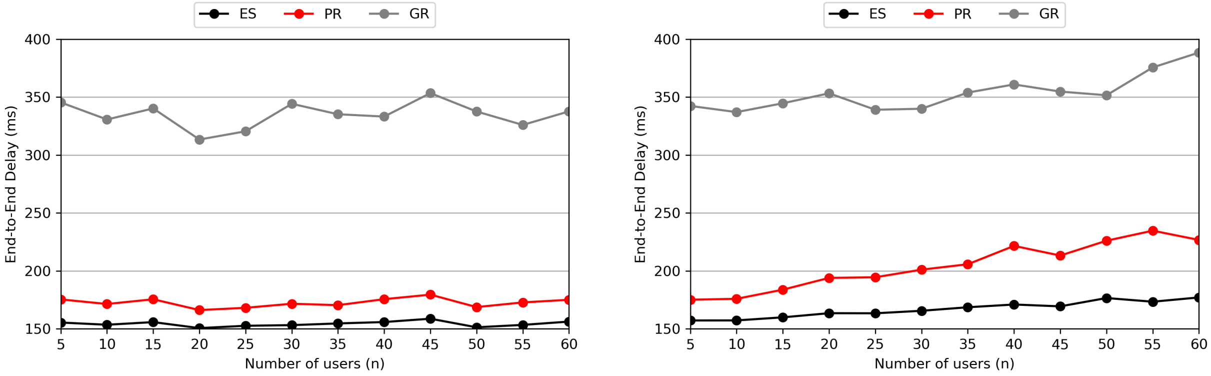
Figure 10. Performance comparison of E2E delay based on computing power.

One drawback of the proposed method is that E2E delay significantly increases in scenarios with low computing power. However, in 3GPP, there are plans to provide regenerative payload services utilizing LEO satellite networks [44]. In other words, by using LEO satellites as base stations, it is possible to deploy VNFs for service provisioning. Based on this development, it is expected that each satellite network will have sufficient computing power, though perhaps not as much as ground stations. Therefore, we believe that our proposed method can outperform other existing techniques in real-world scenarios.

4.6. Analysis of the Scalability

To demonstrate the scalability of our method, we conducted experiments focusing on three aspects: (i) performance comparison based on the number of users, (ii) performance comparison based on the number of SFCs, and (iii) performance comparison based on the number of VNFs. To verify the scalability of our method, we evaluated its performance under two scenarios: our main scenario and a highly loaded scenario. The parameters for the three aspects, n_u, n_c, and n_f, were set to 10, 5, and 20, respectively, in the main scenario and 30, 20, and 40, respectively in the highly loaded scenario. Other parameters were taken from Table 1. The experiments are designed to measure the E2E delay, allowing us to assess how effectively our method scales under different network conditions and parameter settings.

Figure 11 represents the performance comparison based on the number of users. As seen in Figure 11a, within the MHR-scale region we consider, PR shows performance similar to ES even as the number of users increases. This indicates that the E2E delay does not increase significantly because there are relatively few types of SFCs, meaning that the increasing number of users leads to handling similar requests. On the other hand, in Figure 11b, where a higher number of SFCs and VNFs are considered, it is evident that as the number of users increases, the increase in E2E delay is larger compared to ES. This delay increase occurs because low-delay paths become occupied as different types of SFCs are requested.

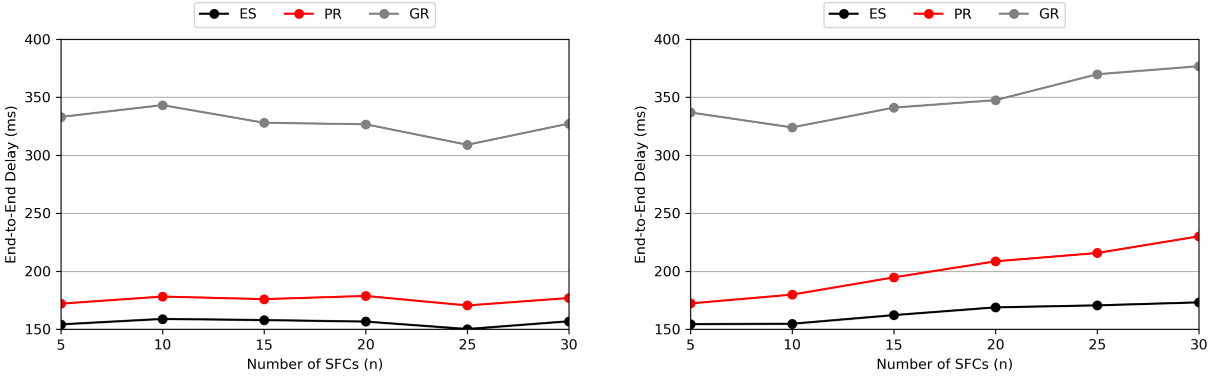


(**a**)

(**b**)

Figure 11. Average E2E delay according to the number of users in (**a**) main scenario and (**b**) highly loaded scenario.

Figure 12 shows the performance comparison based on the number of SFCs. In Figure 12a, even as the variety of SFCs increases, the performance shows little change since the number of users is fixed. Therefore, we can see an increase in delay only up to the point where the number of users matches the number of SFC types, which is ten. In Figure 12b, as the number of users increases, the E2E delay continues to increase. This result is similar to that of the experiment on the number of users, indicating that there is a limitation in VNF placement as various types of SFCs are requested. However, even under a highly loaded scenario, a perfectly individual SFC request situation is unlikely to occur. Thus, we expect that such an increase in E2E delay would not happen within the MHR scenario we consider.



(**a**)

(**b**)

Figure 12. Average E2E delay according to the number of SFCs in (**a**) the main scenario and (**b**) the highly loaded scenario.

In terms of the number of VNFs, we conducted experiments with the number of VNFs in a single SFC randomly set within the range [5, 15] in our typical experimental setup.

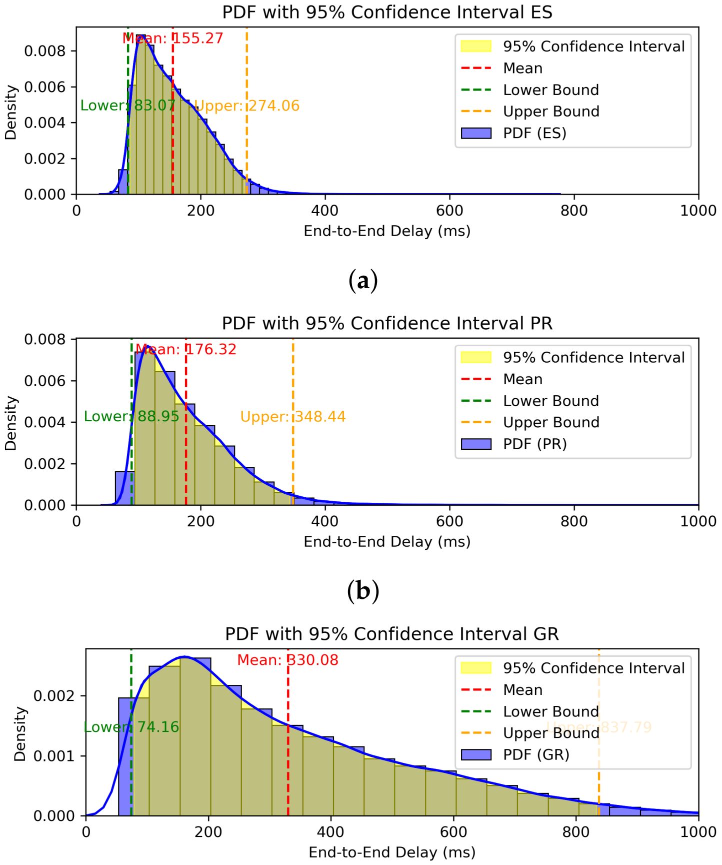
Therefore, examining the number of VNFs per SFC, rather than merely increasing the total number of VNFs, would better reflect scalability. This experiment has been analyzed in detail in Figures 5–7.

From a scalability perspective, we confirmed that the E2E delay increased as the number of users, SFCs, and VNFs increased. Our method focuses on VNF placement and routing path formation within the MHR region. By reducing the problem scope to the MHR, the proposed method is expected to achieve the desired performance in the main scenarios we consider. In other words, we can claim that by solving the problem within the MHR, we have secured scalability for our proposed method.

4.7. Analysis of the Confidence Interval

Since our experiments involve a large number of simulations from various aspects and present averaged results, an analysis of how the E2E delays are distributed is necessary. To derive this distribution, we conducted 10,000 experiments using the parameters in Table 1 and represented the results as a probability density function (PDF). Then, we calculated the overall average E2E delay and derived a 95% confidence interval centered on this average to analyze the distribution of E2E delays.

Figure 13 shows PDFs with a 95% confidence interval of (a) ES, (b) PR, and (c) GR. Comparing the performance of ES and PR, PR shows an increase of approximately 11.93%, 4.32%, and 26.59% in the mean, lower bound, and upper bound, respectively, compared to ES. Although the upper bound shows a substantial increase in E2E delay, most E2E delays are observed to be distributed below the mean value. Therefore, we can conclude that the proposed method exhibits performance similar to that of ES in most cases.



(**a**)

(**b**)

(**c**)

Figure 13. PDFs of E2E delay with 95% confidence interval of (**a**) ES, (**b**) PR, and (**c**) GR.

5. Conclusions

In this paper, we proposed a scheme for VNF placement and routing path formulation in SGIN, aiming to minimize E2E delay. The proposed scheme introduces an MHR method to streamline the identification of candidate base stations for routing paths, thereby reducing computational complexity. By leveraging only the latitude and longitude of the service provider and end user, the MHR effectively narrows down the range of candidate base stations to be considered. Additionally, we developed an algorithm for routing path formation and VNF placement, considering the resource requirements of VNFs, such as computing power and communication link data rates. The routing paths were designed to minimize propagation delays, and VNFs were strategically placed along the paths to reduce both computational and transmission delays. Through simulation-based experiments, we demonstrated the effectiveness of the proposed technique, achieving up to a 97% similarity in performance to the exhaustive search method, with an average similarity of approximately 90%, in terms of minimizing E2E delay while significantly lowering computational complexity. For future work, we identify the need to consider the frequency of SFC invocations. In the current scheme, initially invoked SFCs occupy the optimal routing paths, which may not be ideal when multiple end users are involved. To further optimize E2E delay, routing paths and VNF placements should be dynamically adjusted based on the frequency of SFC invocations. Future research will focus on developing mechanisms that allocate optimal routing paths to frequently invoked SFCs, ensuring efficient VNF placement and minimizing delays across multiple service instances.

Author Contributions: Conceptualization, S.K. and J.P.; methodology, S.K. and J.P.; software, J.P.; validation, S.K., J.P. and J.Y.; formal analysis, S.K., J.P. and S.A.; investigation, S.K. and J.P.; resources, S.K. and J.P.; data curation, S.A.; writing—original draft preparation, S.K., J.P. and S.C.; writing—review and editing, S.K., J.P. and S.C.; visualization, J.P. and Y.J.; supervision, S.K., J.P. and S.C.; project administration, S.C.; funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by an Electronics and Telecommunications Research Institute (ETRI) grant funded by the ICT R&D program of MSIT/IITP (No.2021-0-00847, Development of 3D Spatial Networking Technology/Development of 3D Spatial Satellite Communications Technology), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00704, Development of 3D-NET Core Technology for High-Mobility Vehicular Service), and the Electronics and Telecommunications Research Institute (ETRI) grant funded by the ICT R&D program of MSIT/IITP [RS-2024-00359235, Development of Ground Station Core Technology for Low Earth Orbit Cluster Satellite Communications].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data generated during this current study are available from the authors upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SGIN	Satellite–ground integrated network
QoS	Quality of Service
LEO	Low-Earth orbit
SDN	Software-defined networking
NFV	Network function virtualization
VNF	Virtual Network Function
E2E	End-to-end
SFC	Service Function Chain
MINLP	Mixed integer non-linear programming
MHR	Minimum hop region

References

1. Park, J.; Youn, J.; Oh, J.; Im, J.; Ahn, S.; Kim, S.; Cho, S. Random Access Protocol for Massive Internet of Things Connectivity in Space–Air–Ground-Integrated Networks. *IEEE Internet Things J.* **2023**, *10*, 20442–20457. [CrossRef]
2. Yao, H.; Wang, L.; Wang, X.; Lu, Z.; Liu, Y. The space-terrestrial integrated network: An overview. *IEEE Commun. Mag.* **2018**, *56*, 178–185. [CrossRef]
3. Zhi, L.; Hehao, N.; Yuanzhi, H.; Kang, A.; Xudong, Z.; Zheng, C.; Pei, X. Self-powered absorptive reconfigurable intelligent surfaces for securing satellite-terrestrial integrated networks. *China Commun.* **2024**, *21*, 276–291. [CrossRef]
4. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS-Aided Hybrid Satellite-Terrestrial Relay Networks: Joint Beamforming Design and Optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3717–3724. [CrossRef]
5. Lin, Z.; Lin, M.; Champagne, B.; Zhu, W.P.; Al-Dhahir, N. Secrecy-Energy Efficient Hybrid Beamforming for Satellite-Terrestrial Integrated Networks. *IEEE Trans. Commun.* **2021**, *69*, 6345–6360. [CrossRef]
6. He, Y.; Liu, Y.; Jiang, C.; Zhong, X. Multiobjective Anti-Collision for Massive Access Ranging in MF-TDMA Satellite Communication System. *IEEE Internet Things J.* **2022**, *9*, 14655–14666. [CrossRef]
7. Shafi, M.; Molisch, A.F.; Smith, P.J.; Haustein, T.; Zhu, P.; De Silva, P.; Tufvesson, F.; Benjebbour, A.; Wunder, G. 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1201–1221. [CrossRef]
8. Chen, W.; Lin, X.; Lee, J.; Toskala, A.; Sun, S.; Chiasserini, C.F.; Liu, L. 5G-Advanced Toward 6G: Past, Present, and Future. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1592–1619. [CrossRef]
9. Wang, C.X.; You, X.; Gao, X.; Zhu, X.; Li, Z.; Zhang, C.; Wang, H.; Huang, Y.; Chen, Y.; Haas, H.; et al. On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 905–974. [CrossRef]
10. Kodheli, O.; Lagunas, E.; Maturo, N.; Sharma, S.K.; Shankar, B.; Montoya, J.F.M.; Duncan, J.C.M.; Spano, D.; Chatzinotas, S.; Kisseleff, S.; et al. Satellite Communications in the New Space Era: A Survey and Future Challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 70–109. [CrossRef]
11. Chen, C.; Liao, Z.; Ju, Y.; He, C.; Yu, K.; Wan, S. Hierarchical Domain-Based Multicontroller Deployment Strategy in SDN-Enabled Space–Air–Ground Integrated Network. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 4864–4879. [CrossRef]
12. Yuan, S.; Sun, Y.; Peng, M. Joint Network Function Placement and Routing Optimization in Dynamic Software-Defined Satellite-Terrestrial Integrated Networks. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 5172–5186. [CrossRef]
13. Jia, Z.; Sheng, M.; Li, J.; Zhou, D.; Han, Z. VNF-Based Service Provision in Software Defined LEO Satellite Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 6139–6153. [CrossRef]
14. Guo, J.; Yang, L.; Rincón, D.; Sallent, S.; Chen, Q.; Liu, X. Static Placement and Dynamic Assignment of SDN Controllers in LEO Satellite Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 4975–4988. [CrossRef]
15. Maity, I.; Giambene, G.; Vu, T.X.; Kesha, C.; Chatzinotas, S. Traffic-Aware Resource Management in SDN/NFV-Based Satellite Networks for Remote and Urban Areas. *IEEE Trans. Veh. Technol.* **2024**, *73*, 17400–17415 . [CrossRef]
16. Gao, X.; Liu, R.; Kaushik, A.; Zhang, H. Dynamic Resource Allocation for Virtual Network Function Placement in Satellite Edge Clouds. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 2252–2265. [CrossRef]
17. Li, X.; Tang, F.; Fu, L.; Yu, J.; Chen, L.; Liu, J.; Zhu, Y.; Yang, L.T. Optimized Controller Provisioning in Software-Defined LEO Satellite Networks. *IEEE Trans. Mob. Comput.* **2023**, *22*, 4850–4864. [CrossRef]
18. He, J.; Cheng, N.; Yin, Z.; Zhou, H.; Zhou, C.; Aldubaikhy, K.; Alqasir, A.; Shen, X. Load-Aware Network Resource Orchestration in LEO Satellite Network: A GAT-Based Approach. *IEEE Internet Things J.* **2024**, *11*, 15969–15984. [CrossRef]
19. Han, C.; Li, X.; Ji, H.; Zhang, H. Adaptive Online Service Function Chain Deployment in Large-scale LEO Satellite Networks. In Proceedings of the 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Toronto, ON, Canada, 5–8 September 2023; pp. 1–6. [CrossRef]
20. Xia, Q.; Wang, G.; Xu, Z.; Liang, W.; Xu, Z. Efficient Algorithms for Service Chaining in NFV-Enabled Satellite Edge Networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 5677–5694. [CrossRef]
21. Wang, G.; Zhou, S.; Zhang, S.; Niu, Z.; Shen, X. SFC-Based Service Provisioning for Reconfigurable Space-Air-Ground Integrated Networks. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1478–1489. [CrossRef]
22. Gao, X.; Liu, R.; Kaushik, A. Service Chaining Placement Based on Satellite Mission Planning in Ground Station Networks. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 3049–3063. [CrossRef]
23. Li, J.; Shi, W.; Wu, H.; Zhang, S.; Shen, X. Cost-Aware Dynamic SFC Mapping and Scheduling in SDN/NFV-Enabled Space–Air–Ground-Integrated Networks for Internet of Vehicles. *IEEE Internet Things J.* **2022**, *9*, 5824–5838. [CrossRef]
24. Liu, Y.; Wang, L.; Lu, Z.; Du, K.; Shou, G. A Stateless Design of Satellite–Terrestrial Integrated Core Network and Its Deployment Strategy. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 953–966. [CrossRef]
25. Tang, Q.; Fei, Z.; Li, B.; Han, Z. Computation Offloading in LEO Satellite Networks with Hybrid Cloud and Edge Computing. *IEEE Internet Things J.* **2021**, *8*, 9164–9176. [CrossRef]
26. Zhang, Y.; Chen, C.; Liu, L.; Lan, D.; Jiang, H.; Wan, S. Aerial Edge Computing on Orbit: A Task Offloading and Allocation Scheme. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 275–285. [CrossRef]
27. Cao, X.; Yang, B.; Shen, Y.; Yuen, C.; Zhang, Y.; Han, Z.; Poor, H.V.; Hanzo, L. Edge-Assisted Multi-Layer Offloading Optimization of LEO Satellite-Terrestrial Integrated Networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 381–398. [CrossRef]
28. He, P.; Hu, J.; Fan, X.; Wu, D.; Wang, R.; Cui, Y. Load-Balanced Collaborative Offloading for LEO Satellite Networks. *IEEE Internet Things J.* **2023**, *10*, 19075–19086. [CrossRef]

29. Kim, T.; Kwak, J.; Choi, J.P. Satellite Edge Computing Architecture and Network Slice Scheduling for IoT Support. *IEEE Internet Things J.* **2022**, *9*, 14938–14951. [CrossRef]
30. Tang, Q.; Fei, Z.; Li, B.; Yu, H.; Cui, Q.; Zhang, J.; Han, Z. Stochastic Computation Offloading for LEO Satellite Edge Computing Networks: A Learning-Based Approach. *IEEE Internet Things J.* **2024**, *11*, 5638–5652. [CrossRef]
31. Wang, D.; Wang, W.; Kang, Y.; Han, Z. Distributed Data Offloading in Ultra-Dense LEO Satellite Networks: A Stackelberg Mean-Field Game Approach. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 112–127. [CrossRef]
32. Petrosino, A.; Piro, G.; Grieco, L.A.; Boggia, G. On the Optimal Deployment of Virtual Network Functions in Non-Terrestrial Segments. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 4831–4845. [CrossRef]
33. Qin, X.; Ma, T.; Tang, Z.; Zhang, X.; Zhou, H.; Zhao, L. Service-Aware Resource Orchestration in Ultra-Dense LEO Satellite-Terrestrial Integrated 6G: A Service Function Chain Approach. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 6003–6017. [CrossRef]
34. Zhang, P.; Zhang, Y.; Kumar, N.; Guizani, M. Dynamic SFC Embedding Algorithm Assisted by Federated Learning in Space–Air–Ground-Integrated Network Resource Allocation Scenario. *IEEE Internet Things J.* **2023**, *10*, 9308–9318. [CrossRef]
35. He, J.; Cheng, N.; Yin, Z.; Zhou, C.; Zhou, H.; Quan, W.; Lin, X.H. Service-Oriented Network Resource Orchestration in Space-Air-Ground Integrated Network. *IEEE Trans. Veh. Technol.* **2024**, *73*, 1162–1174. [CrossRef]
36. Hu, M.; Yang, R.; Hu, Y.; Cai, C.; Dong, Y.; Deng, T.; Peng, K. QoS-Aware Software-Defined Multicast in LEO Satellite Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 5307–5317. [CrossRef]
37. Hu, M.; Li, J.; Cai, C.; Deng, T.; Xu, W.; Dong, Y. Software Defined Multicast for Large-Scale Multi-Layer LEO Satellite Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 2119–2130. [CrossRef]
38. Hu, M.; Xiao, M.; Hu, Y.; Cai, C.; Deng, T.; Peng, K. Software Defined Multicast Using Segment Routing in LEO Satellite Networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 835–849. [CrossRef]
39. Hu, M.; Xiao, M.; Xu, W.; Deng, T.; Dong, Y.; Peng, K. Traffic Engineering for Software-Defined LEO Constellations. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 5090–5103. [CrossRef]
40. Wang, L.; Mao, W.; Zhao, J.; Xu, Y. DDQP: A Double Deep Q-Learning Approach to Online Fault-Tolerant SFC Placement. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 118–132. [CrossRef]
41. Yuan, J.; Chen, P.; Liu, Q.; Li, H. A load-balanced on-demand routing for LEO satellite networks. *J. Netw.* **2014**, *9*, 3305. [CrossRef]
42. Im, J.; Youn, J.; Kim, S.; Park, J.; Lee, S.; Kwon, Y.; Cho, S. Distributed Detour Routing Scheme for Link Failure with Minimized Overhead in LEO Satellite Networks. *Sensors* **2023**, *23*, 9590. [CrossRef] [PubMed]
43. Qin, P.; Wang, M.; Zhao, X.; Geng, S. Content Service Oriented Resource Allocation for Space–Air–Ground Integrated 6G Networks: A Three-Sided Cyclic Matching Approach. *IEEE Internet Things J.* **2023**, *10*, 828–839. [CrossRef]
44. 3GPP. *Solutions for NR to Support Non-Terrestrial Networks (NTN)*, Version 16.2.0; Technical Report (TR) 38.821, 3rd Generation Partnership Project (3GPP); European Telecommunications Standards Institute (ETSI): Sophia Antipolis, France, 2023.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.