Dynamic Resource Management for Neighbor-based VNF Placement in Decentralized Satellite Networks

Xiangqiang Gao[1], Rongke Liu[1], Aryan Kaushik[2], John Thompson[3], Hangyu Zhang[1] and Yuepu Ma[1]
[1]School of Electronic and Information Engineering, Beihang University, Beijing, China
[2]School of Engineering and Informatics, University of Sussex, Brighton, United Kingdom
[3]Institute for Digital Communications, University of Edinburgh, Edinburgh, United Kingdom
Email: {xggao, rongke_liu, zhanghangyu, yuepu_ma}@buaa.edu.cn, aryan.kaushik@sussex.ac.uk, j.s.thompson@ed.ac.uk

***Abstract*—By introducing software-defined networking (SDN) and network function virtualization (NFV), low-earth-orbit (LEO) satellite networks can facilitate virtual network function (VNF) placement, which will provide computing services for satellite applications on-demand. In this paper, we study the VNF placement problem in a decentralized LEO satellite network due to the requirements for real-time processing and network resilience, where our aim is to jointly optimize end-to-end service delay and network bandwidth cost in a dynamic environment. To this end, a decentralized LEO satellite network architecture is first implemented for resource management by establishing the neighboring sub-network for each satellite. Then we formulate the VNF placement problem as an integer non-linear programming problem with multiple constraints of network resources and service requirements. A neighbor-based VNF placement (N-VNFP) approach is proposed to address the optimization problem. Finally, we conduct the experiments to evaluate the performance of the proposed N-VNFP approach in a Walker constellation with 66 LEO satellites. The simulation results show that the proposed N-VNFP approach provides an effective solution for resource management in a decentralized LEO satellite network and also outperforms the two centralized baselines, i.e., Viterbi and Greedy, in terms of end-to-end service delay and network bandwidth cost.**

***Index Terms*—Satellite network, virtual network function, resource management, distributed optimization.**

I. Introduction

There is currently a rapid increase in satellite services and applications, e.g., for remote sensing, environment monitoring, target recognition, etc. Low-earth-orbit (LEO) satellite networks are required to provide service provisioning in a flexible and efficient way, as a large amount of data is produced by these satellite services and applications. This needs to be processed, stored, and transmitted in real-time [1]. However, flexible service provisioning can not be properly supported in existing proprietary satellite networks.

Software-defined networking (SDN) and network function virtualization (NFV) have emerged as new paradigms in decoupling software and hardware, enabling network resources to be allocated on-demand [2]. By integrating SDN and NFV into LEO satellite networks, flexible service provisioning for satellite applications is possible [3]. Note that on-board resources in LEO satellite networks, such as computing, storage, and bandwidth, are severely limited compared to the terrestrial networks, allocating on-board resources in SDN/NFV-enabled LEO satellite networks has been a challenging problem [4].

Some existing research work has focused on the resource allocation problem in SDN/NFV-enabled LEO satellite networks [5]–[7]. The authors in [5] discussed a dynamic controller placement in SDN-enabled LEO satellite networks, where the traffic demands can change dynamically. In [6], the authors proposed a dynamic resource allocation architecture and implemented an advanced K-means algorithm to address the problem in satellite edge computing. A virtual network function (VNF) embedding and data routing joint optimization problem is considered in software-defined satellite networks in [7]. However, these existing work considered the resource management problem in only centralized SDN/NFV satellite networks, where one or more controllers are deployed to manage satellite network resources for satellite applications. Considering the mobility of satellites, it is difficult for centralized architectures to manage on-board resources efficiently in large-scale satellite networks. Moreover, real-time processing requirements for satellite applications can not be guaranteed due to the high transmission delay in typical satellite networks.

In this paper, we investigate the resource management problem for VNF placement in a decentralized LEO satellite network [8], where no centralized controllers are used to manage network resources. Each satellite is viewed as an agent to manage the neighboring sub-network resources and provide computing services for satellite applications. Here, the neighboring sub-network for each satellite is composed of nearby satellites [9]. We first implement a satellite network architecture based on the neighboring sub-networks for resource management in a decentralized satellite network. Then we formulate the VNF placement problem as an integer non-linear programming problem with multiple constraints to jointly optimize end-to-end service delay and network bandwidth cost. Due to the assumption that satellite applications should be processed by the satellite and its neighbors as much as possible to guarantee the service quality [10], we propose a neighbor-based VNF placement (N-VNFP) approach, which can provide computing services for satellite applications in parallel and improve the real-time performance of VNF placement, to manage the network resources in a decentralized LEO satellite network. Finally, we evaluate the performance of the proposed N-VNFP approach compared with Viterbi [11] and Greedy [12] in a Walker constellation with 66 LEO satellites.

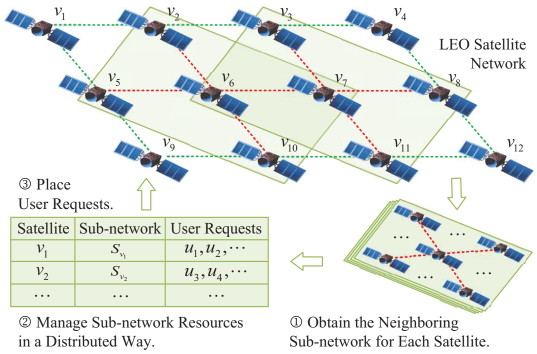


Fig. 1. Decentralized LEO satellite network architecture.

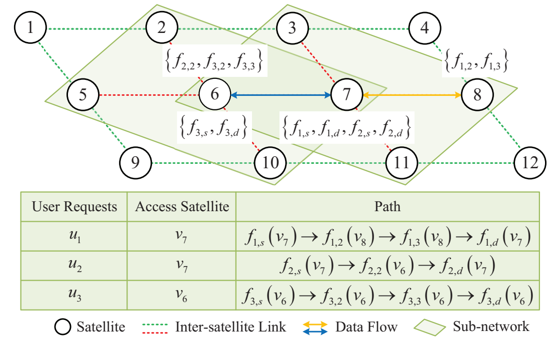


Fig. 2. Example of VNF placement in a decentralized satellite network.

The reminder of this paper is organized as follows. Section II introduces the system model. In Section III, we formulate the VNF placement problem and propose the N-VNFP approach to address it. In Section IV, we conduct the experiments to evaluate the performance of the proposed N-VNFP approach. Finally, we provide the conclusion in Section V.

II. System Model

A. Decentralized LEO Satellite Network

1) Network Architecture: The proposed architecture of a decentralized LEO satellite network is shown in Fig. 1. The satellite network is made up of LEO satellites and each satellite has four inter-satellite links with its neighbors, where two inter-satellite links are from satellites in the same orbit and the other two links are from satellites in different orbits [13]. The neighboring sub-network of each satellite, which is shown as step 1 in Fig. 1, can be established through multi-hop communication between satellites. Each satellite has the autonomous ability to manage the neighboring sub-network resources and place the VNFs for satellite applications in the neighboring sub-network. Each satellite application can be viewed as a user request and consists of multiple VNFs in a specific order. When a user request accesses the network through one satellite, the satellite will be considered as the access satellite for the user request. In the proposed decentralized satellite network, all the satellites can manage their neighboring sub-network resources and provide computing services for user requests in parallel, as shown in step 2 in Fig. 1. Then we can deploy user requests by solutions obtained in step 2 to the satellite network in a distributed and cost-efficient way, as shown in step 3 in Fig. 1. Note that the satellite network is built based on a time-evolving method [9], where the satellite network topology is unchanged in a time slot and can be varying over different time slots.

2) Network Model: The satellite network is indicated as a graph $G(V, E)$, where V and E are indicated the set of satellites and the set of links, respectively. For each satellite, two physical resources of central processing unit (CPU) and memory are considered as $R = \{CPU, Memory\}$, and the r-th resource capacity for satellite v is denoted as $C_v^r, r \in R$. For link e, we denote the bandwidth capacity as B_e and the transmission delay as t_e. The hops of neighboring sub-network S_v for satellite v can be denoted as h, where the distance between two neighboring satellites is indicated as one hop.

B. User Requests

The set of user requests is indicated as U with M user requests and user request u is indicated as a directed acyclic graph $G(F_u, H_u)$. The set of VNFs is represented as $F_u = \{f_{u,1} = f_{u,s}, f_{u,2}, \cdots, f_{u,|F_u|} = f_{u,d}\}$, where $|F_u|$ indicates the number of VNFs in F_u, $f_{u,s}$ and $f_{u,d}$ are the source and the destination of user request u to route traffic flow, respectively. For VNF $f_{u,i}$, we indicate the r-th required resources as $c_{u,i}^r, r \in R$, and the computing time as $t_{u,i}$. The set of edges for user request u is denoted as H_u. We indicate the edge between two adjacent VNFs f_{u,i_1} and f_{u,i_2} as $h_u^{i_1,i_2}$ and the bandwidth requirements as $b_u^{i_1,i_2}$. The maximum acceptable delay for user request u is denoted as t_u^{max}.

C. VNF Placement in a Decentralized Satellite Network

Similar to the existing work related to edge computing [7], we provide computing services for user requests in a quasi-static scenario. In each time slot, user requests can randomly arrive following a Poisson distribution and ask for computing services from a decentralized LEO satellite network. The computing service can be provided for each user request by the neighboring sub-network of the access satellite and all user requests can be deployed to the satellite network in parallel. We assume that the source and the destination for each user request are both located at the access satellite. The service requirements of user requests can be also guaranteed in the running time. When user requests are deployed to the satellite network, the adjacent VNFs from a user request will be deployed on the same satellite as much as possible to reduce the deployment cost in terms of end-to-end service delay and network bandwidth cost. Note that the required resources of user requests may exceed the resource capacity of a satellite or an inter-satellite link as multiple user requests from different neighbouring sub-networks can ask for the available resources of satellites and links at similar times, which can bring a potential resource conflict. This potential resource conflict can be addressed by a first-come first-served method, which will be discussed in detail later.

Fig. 2 shows an example of placing VNFs in a decentralized satellite network. There are 12 LEO satellites and the neighboring sub-networks for satellites v_6 and v_7 are established with $h = 1$, respectively. Three user requests are represented as $u_1 = \{f_{1,s}, f_{1,2}, f_{1,3}, f_{1,d}\}$, $u_2 = \{f_{2,s}, f_{2,2}, f_{2,d}\}$, and $u_3 = \{f_{3,s}, f_{3,2}, f_{3,3}, f_{3,d}\}$, respectively. The access satellite for user requests u_1 and u_2 is satellite v_7, and then the neighboring sub-network for satellite v_7 will provide computing services for the two user requests. Due to the limited resources of satellite v_7, VNFs $f_{1,2}$ and $f_{1,3}$ are deployed to satellite v_8 and VNF $f_{2,2}$ is deployed to satellite v_6. The access satellite for user request u_3 is satellite v_6, which can provide sufficient available network resources to deploy $f_{3,2}$ and $f_{3,3}$. The VNF placement strategies for three user requests are shown in Fig. 2.

III. Problem Formulation and Proposed Approach

A. Problem Formulation

Considering the real-time service requirements and the limited network bandwidth resources, we investigate the VNF placement problem to jointly optimize end-to-end service delay and network bandwidth cost for user requests.

We first denote a binary variable $x_u = \{0, 1\}$ to indicate whether user request u is deployed to the satellite network. Here, $x_u = 1$ if user request u is implemented by the neighboring satellites, otherwise $x_u = 0$.

When user request u is deployed to the satellite network, we use another binary variable $y_{u,i}^{v} = \{0, 1\}$ to indicate whether VNF $f_{u,i}$ is deployed to satellite v. Again, $y_{u,i}^{v} = 1$ if VNF $f_{u,i}$ is deployed to satellite v, otherwise $y_{u,i}^{v} = 0$.

When two adjacent VNFs for a user request are deployed to different satellites, one available path between the two satellites will be used to route the traffic flow between the two adjacent VNFs. We assume that the d shortest paths between two satellites can be computed in advance, where the path set for satellites v_1 and v_2 can be indicated as $P_{v_1}^{v_2}$. We use a binary variable $z_{u,p}^{i_1,i_2} = \{0, 1\}$ to indicate whether path p, $p \in P_{v_1}^{v_2}$, is used by edge $h_u^{i_1,i_2}$ to route the traffic flow. $z_{u,p}^{i_1,i_2} = 1$ if path p is used by edge $h_u^{i_1,i_2}$, otherwise $z_{u,p}^{i_1,i_2} = 0$.

We also use a binary variable $q_e^p = \{0, 1\}$ to indicate whether link e is used by path p. If link e is used by path p then $q_e^p = 1$, otherwise $q_e^p = 0$.

For the VNF placement problem, each VNF for a user request can be deployed to one and only one satellite. We assume that the access satellite for user request u is satellite v, then the VNF deployment constraint can be described as

$$\sum_{\bar{v} \in S_v} y_{u,i}^{\bar{v}} = 1, \forall f_{u,i} \in F_u. \tag{1}$$

Given the limited network resources, when we deploy the VNFs for user requests to a satellite we need to guarantee that the resource requirements of user requests should not exceed the resource capacity. The resource constraint for satellite v can be described as

$$\sum_{u \in U} \sum_{f_{u,i} \in F_u} y_{u,i}^{v} \cdot c_{u,i}^{r} \leq C_v^r, \forall r \in R. \tag{2}$$

When two adjacent VNFs for a user request are deployed to different satellites, it is guaranteed that one of available paths between the two satellites will be used by the edge to route the traffic flow between the two adjacent VNFs. We also assume that the access satellite for user request u is satellite v, adjacent VNFs f_{u,i_1} and f_{u,i_2} are deployed to satellites v_1 and v_2, respectively. The path selection constraint can be indicated as

$$\sum_{v_1 \in S_v} \sum_{\substack{v_2 \in S_v, \\ v_1 \neq v_2}} \left| y_{u,i_1}^{v_1} \cdot y_{u,i_2}^{v_2} - \sum_{p \in P_{v_1}^{v_2}} z_{u,p}^{i_1,i_2} \right| = 0. \tag{3}$$

We also guarantee that the bandwidth requirements for user requests should not exceed the bandwidth capacity of link e when user requests are deployed to the satellite network. The bandwidth resource constraint for link e can be indicated as

$$\sum_{u \in U} \sum_{h_u^{i_1,i_2} \in H_u} \sum_{v_1 \in V} \sum_{\substack{v_1 \in V, \\ v_1 \neq v_2}} \sum_{p \in P_{v_1}^{v_2}} y_{u,i_1}^{v_1} \cdot y_{u,i_2}^{v_2} \cdot z_{u,p}^{i_1,i_2} \cdot q_e^p \cdot b_u^{i_1,i_2} \leq B_e. \tag{4}$$

Furthermore, when user request u is deployed to the satellite network, we need to ensure that the service delay cannot exceed the maximum acceptable delay. We use t_u to indicate the service delay for user request u. The service delay constraint can be indicated as

$$t_u = \sum_{f_{u,i} \in F_u} t_{u,i} + t_u^{tr} \leq t_u^{max}, \tag{5}$$

where transmission delay t_u^{tr} can be described as

$$t_u^{tr} = \sum_{h_u^{i_1,i_2} \in H_u} \sum_{v_1 \in S_v} \sum_{\substack{v_2 \in S_v, \\ v_1 \neq v_2}} \sum_{p \in P_{v_1}^{v_2}} \sum_{e \in p} y_{u,i_1}^{v_1} \cdot y_{u,i_2}^{v_2} \cdot z_{u,p}^{i_1,i_2} \cdot t_e. \tag{6}$$

For the VNF placement problem, end-to-end service delay C_{delay} and network bandwidth cost C_{bw} for user requests can be described as

$$\begin{cases} C_{delay} = \frac{1}{\sum_{u \in U} x_u} \cdot \sum_{u \in U} x_u \cdot t_u \\ C_{bw} = \frac{1}{\sum_{u \in U} x_u} \cdot \sum_{u \in U} x_u \cdot c_u^{bw}, \end{cases} \tag{7}$$

where the network bandwidth cost of user request u can be indicated as

$$c_u^{bw} = \sum_{h_u^{i_1,i_2} \in H_u} \sum_{v_1 \in S_v} \sum_{\substack{v_2 \in S_v, \\ v_1 \neq v_2}} \sum_{p \in P_{v_1}^{v_2}} \sum_{e \in p} y_{u,i_1}^{v_1} \cdot y_{u,i_2}^{v_2} \cdot z_{u,p}^{i_1,i_2} \cdot b_u^{i_1,i_2}. \tag{8}$$

Therefore, the optimization problem for VNF placement can be formulated as

$$\begin{aligned} min \quad & C_{delay}, C_{bw} \\ s.t. \quad & (1) - (5). \end{aligned} \tag{9}$$

For the optimization problem in equation (9), we consider the end-to-end service delay as the primary optimization objective and the network bandwidth cost as the secondary optimization objective. Here the primary optimization objective has a high priority in finding VNF placement strategies when user requests are deployed to the satellite network.

Algorithm 1 Neighbor-based VNF Placement.

Input: User requests U;
Output: $a^* = \{a_u^* | u \in U\}$;

Initialize: $\forall u \in U, a_u = a_u^* = \varnothing, U_{next} = \varnothing$;
while $U \neq \varnothing$ **do**
  Update the neighboring sub-network resource state for each satellite;
  for each satellite $v \in V$ in parallel **do**
    for each user request u in access satellite v **do**
      Search VNF placement strategy a_u by Viterbi in the neighboring sub-network;
    end for
  end for
  for each user request $u \in U$ in parallel **do**
    if $a_u \neq \varnothing$ and there is no resource conflict **then**
      Deploy user request u by a_u and $a_u^* \leftarrow a_u$;
    else if $a_u \neq \varnothing$ and there is resource conflict **then**
      Initialize $a_u = \varnothing$ and $U_{next} = U_{next} \cup \{u\}$;
    end if
  end for
  $U \leftarrow U_{next}$ and $U_{next} = \varnothing$;
end while
return a^*;

Algorithm 2 Viterbi Algorithm.

Input: user request u, access satellite v;
Output: a_u;

Initialize: $a_u = \varnothing, A = \{a_{u,s}\}$;
Obtain the topology Γ_u for the VNFs except $f_{u,s}$;
for each $f_{u,i} \in \Gamma_u$ **do**
  $A_{next} = \varnothing$;
  for each $a \in A$ **do**
    Update the neighboring sub-network resource state;
    for each $\bar{v} \in S_v$ **do**
      Calculate the required resources for the satellite;
      Sort the paths by delay in an ascending order;
      for each candidate path from the path set **do**
        if all resource requirements are satisfied **then**
          Obtain the VNF placement strategy $a_{u,i}$;
          $A_{next} = A_{next} \cup \{a_{u,i}\}$ and break;
        end if
      end for
    end for
  end for
  Sort A_{next} by deployment cost in an ascending order;
  $A \leftarrow A_{next}[: B]$;
end for
if $A \neq \varnothing$ **then**
  Obtain strategy a_u with minimum deployment cost of service delay and network bandwidth;
end if
return a_u;

B. Proposed Approach

As the VNF placement problem is proved to be NP-hard in [11], [12], we propose the neighbor-based VNF placement approach using the Viterbi algorithm to tackle the VNF placement problem in a decentralized LEO satellite network. For the proposed N-VNFP approach, each satellite can only obtain the neighboring sub-network resource information to manage the neighboring sub-network resources and provide computing services for user requests in the neighboring sub-network. All satellites can make the VNF placement strategies for user requests in parallel, where the VNF placement strategy for each user request is implemented by the Viterbi algorithm exploiting the neighboring sub-network of the access satellite. In that case, satellite resource capacities may be less than the resource requirements of user requests as user requests from different neighboring sub-networks are deployed to the satellite network in parallel, then we consider that there may be a resource conflict among these user requests.

For a user request, the VNF placement strategy can be made by the access satellite and the VNFs are usually deployed to multiple satellites due to the limited resources of satellites. In order to place the VNFs and build the service chain, the VNF placement strategy for a user request needs to be shared among the related satellites. Each satellite has a first-in first-out sequence to receive the VNF placement strategies from other satellites and can provide computing resources for VNF placement via a first-come first-served method to address the potential resource conflict problem. That is, each satellite has priority to provide computing services for user requests by the VNF placement strategies that come first. When the resource requirements of the VNFs are more than the resource capacity, we consider that these user requests will fail to deploy to the satellite network in the current iteration. Under these circumstances, the access satellites will update the resource information about their neighboring sub-networks and then re-deploy the VNFs from these failed user requests. If all user requests are either deployed to the satellite network or they fail to deploy to their neighboring sub-networks due to the resource limitation, the iteration will terminate. The proposed N-VNFP approach is shown in Algorithm 1, which can be run on all satellites in parallel.

For each user request, the Viterbi algorithm [11] is used to deploy the VNFs and choose the routing path in the neighboring sub-network, where we consider the end-to-end service delay optimization as the primary objective and the network bandwidth cost optimization as the secondary objective. The Viterbi algorithm can be implemented based on a multi-stage graph for the states and their relationships. We assume that each placement strategy for a VNF is viewed as a possible state, all possible states from the same VNF can make up a stage, and the number of stages is equal to the number of the VNFs for a user request. However, there is a deployment cost in terms of CPU, memory, and computing delay when we deploy the VNFs to the satellites. The weighted edge between two possible states from two adjacent stages indicates a path cost, such as the transmission cost and the bandwidth cost. For the Viterbi search tree, we calculate the cumulative cost, which includes the VNF deployment cost and the path cost, and remain B VNF placement strategies with minimum deployment costs in the current stage into the next stage. When the Viterbi search procedure is over, then the most likely path with minimum cumulative cost can be an approximate VNF placement strategy for the user request. The Viterbi algorithm is shown in Algorithm 2, which can be run on each satellite to manage the neighboring sub-network resources.

For the Viterbi algorithm, we can indicate the computational complexity as $O(dB|S_v||F|)$, where the number of satellites in a neighboring sub-network is $|S_v|$ and the average number of VNFs for a user request is $|F|$. For the proposed N-VNFP approach, only one user request can be deployed to the satellite network for each iteration in the worst case, then the maximum number of iterations is $\frac{M^2+M}{2}$. The computational complexity of the proposed N-VNFP approach can be indicated as $O(M^2dB|S_v||F|)$. We can find that the computational complexity of the proposed N-VNFP approach will increase as the search tree width increases as well as the number of candidate paths. Due to the real-time performance of the proposed N-VNFP approach, there is a tradeoff between computational complexity and solution performance.

IV. Simulation Results

To evaluate the performance of the proposed N-VNFP approach in a decentralized LEO satellite network, we conduct the experiments in a Walker constellation [14] with 66 LEO satellites, where the number of orbits is 6, each orbit includes 11 LEO satellites, and the seed satellite parameters for altitude, inclination, and right ascension of ascending node are 780 km, 86 degree, and 15 degree, respectively, and others are viewed as zero. Each satellite has 96 vCPUs and 112 GB memory, the bandwidth capacity of each link is 100 Mbps. For each user request, the number of the VNFs follows a truncated power-law distribution with exponent 2, where the minimum value is 2 and the maximum value is 7 [15]. The required resources for each VNF except the source and the destination follow uniform integer probability distributions as follows $[1, 2]$ vCPUs, $[2, 4]$ GB, and $[10, 30]$ milliseconds and the required bandwidths for each edge can be randomly generated by $[1, 4]$ Mbps. We assume that the running period for each user request follows the exponential distribution, where the mean number of time slots is 3 and the maximum acceptable delay can guarantee the VNF deployment in the neighboring sub-network. The number of candidate paths between two satellites is 8 and the width of the Viterbi search tree is 4 [8]. Each experiment is run over a period of 30 time slots and repeated for 10 times, then we obtain the average results.

For the proposed N-VNFP approach, the neighboring sub-network scale has an impact on the VNF placement performance, where the neighboring sub-network for each satellite is established by the hops. Therefore, we conduct the experiments to obtain the simulation results of the proposed N-VNFP approach in four neighboring sub-networks with

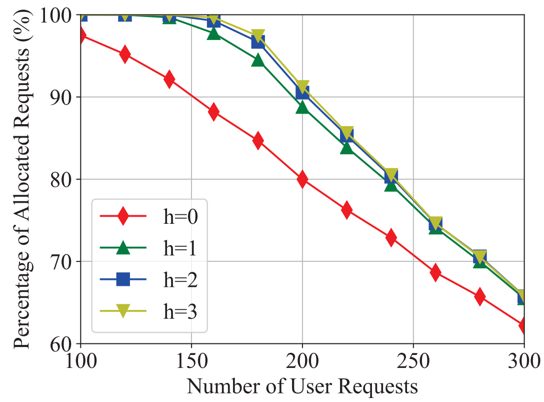
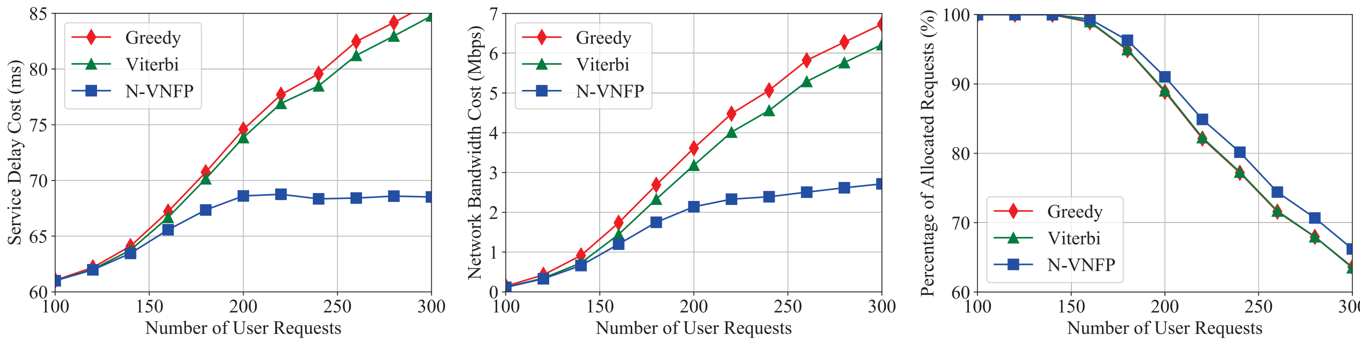


Fig. 3. Percentages of allocated requests in neighboring sub-networks.

different hops, i.e., $h = \{0, 1, 2, 3\}$. Fig. 3 shows the average percentage results of allocated user requests in the four neighboring sub-networks. As the increase in the number of hops, the neighboring sub-network for each satellite can provide more available network resources for user requests, then the percentage of allocated user requests for the proposed N-VNFP approach can correspondingly increase. Compared with the case of $h = 0$, the average performance improvement of the proposed N-VNFP approach is 7.93% for $h = 1$, 9.01% for $h = 2$, and 9.26% for $h = 3$. Note that the complexity of resource management also increases in a decentralized satellite network as the number of hops increases. Therefore, there is a tradeoff between solution performance and network design complexity for the proposed N-VNFP approach. Based on the above discussion, we can find that the performance difference between the cases of $h = 2$ and 3 is relatively close and then the number of sub-network hops can be considered as $h = 2$.

In order to meaningfully evaluate the performance of the proposed N-VNFP approach in a decentralized satellite network, two centralized VNF placement approaches of Viterbi [11] and Greedy [12] are used as the baseline cases, which deploy user requests to the satellite network one by one. We conduct the experiments for the neighboring sub-network with $h = 2$ and obtain simulation results for different user requests in terms of end-to-end service delay, network bandwidth cost, and percentage of allocated user requests, as shown in Fig. 4.

Fig. 4(a) illustrates the average end-to-end service delay results for different user requests. We can observe that the end-to-end service delay results for Greedy, Viterbi, and N-VNFP are close to each other when the number of user requests is small. However, the performance difference for Greedy, Viterbi, and N-VNFP will be obvious as the number of user requests increases, where the proposed N-VNFP approach will outperform the two baselines of Greedy and Viterbi. That is due to the fact that the available network resources are gradually decreasing as the number of user requests increases. For the proposed N-VNFP approach, when the available network resources cannot satisfy with the resource requirements of user requests, some user requests will fail to deploy as the resource limitation. Then these failed user requests can be



(a) End-to-end service delay

(b) Network bandwidth cost

(c) Percentage of allocated user requests

Fig. 4. Simulation results for different user requests with Greedy, Viterbi, and proposed N-VNFP.

re-deployed to the satellite network after updating these sub-network resource states, which can improve the exploration ability of solution space. However, Greedy and Viterbi are used to deploy user requests one-by-one. On average, the performance improvement of the proposed N-VNFP approach for all cases in Fig. 4(a) is 9.78% over Greedy and 8.89% over Viterbi. The average bandwidth cost results for different user requests are shown in Fig. 4(b). Similar to the results in Fig. 4(a), the average bandwidth costs obtained by the proposed N-VNFP approach are better than that of Greedy and Viterbi. The proposed N-VNFP approach reduces the average bandwidth cost by 50.50% over Greedy and 44.82% over Viterbi. We also provide the percentage results of allocated user requests in Fig. 4(c). Considering that the proposed N-VNFP approach performs better than the two baselines of Greedy and Viterbi for reducing the deployment costs of user requests, more user requests can be deployed to the satellite network by the proposed N-VNFP approach compared with Greedy and Viterbi. The average percentage of allocated user requests obtained by the proposed N-VNFP approach is 1.89% over Greedy and 1.83% over Viterbi, respectively.

Based on the above discussion, it is demonstrated that the proposed N-VNFP approach is effective and efficient to allocate the available network resources and place the VNFs for user requests in decentralized satellite networks, where the proposed N-VNFP approach also has good dynamic scalability and can be applied in different scale satellite networks.

V. Conclusion

In this paper, we discussed the dynamic resource management for VNF placement in a decentralized LEO satellite network, where SDN and NFV are introduced to provide flexible and efficient service provisioning. We implemented a decentralized satellite network architecture by the neighboring sub-network to manage network resources in a distributed way. Then we formulated the VNF placement problem as an integer non-linear programming problem to jointly optimize end-to-end service delay and bandwidth cost. We proposed the N-VNFP approach to address the problem. The simulation results in a Walker constellation with 66 LEO satellites demonstrated the effectiveness of the proposed N-VNFP approach for deploying user requests in a decentralized satellite network. Compared with Greedy and Viterbi, the proposed N-VNFP approach can reduce the end-to-end service delay by up to 10%, save the bandwidth cost by around 50%, and improve the percentage of allocated user requests by up to 2%, respectively.

References

[1] S. Xu *et al.*, "Software-defined next-generation satellite networks: Architecture, challenges, and solutions," *IEEE Access*, vol. 6, pp. 4027–4041, 2018.

[2] J. Gil Herrera *et al.*, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Serv. Manag.*, vol. 13, no. 3, pp. 518–532, 2016.

[3] L. Bertaux *et al.*, "Software defined networking and virtualization for broadband satellite networks," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 54–60, 2015.

[4] G. Wang *et al.*, "SFC-based service provisioning for reconfigurable space-air-ground integrated networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1478–1489, 2020.

[5] A. Papa *et al.*, "Design and evaluation of reconfigurable SDN LEO constellations," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 3, pp. 1432–1445, 2020.

[6] F. Wang *et al.*, "Fine-grained resource management for edge computing satellite networks," in *Proc. IEEE Global Commun. Conf.*, Waikoloa, USA, Dec. 2019, pp. 1–6.

[7] Z. Jia *et al.*, "Joint optimization of VNF deployment and routing in software defined satellite networks," in *Proc. IEEE Veh. Technol. Conf.*, Chicago, USA, Aug. 2018, pp. 1–5.

[8] X. Gao *et al.*, "Virtual network function placement in satellite edge computing with a potential game approach," *IEEE Trans. Netw. Serv. Manag.*, doi: 10.1109/TNSM.2022.3141165, 2020.

[9] Q. He *et al.*, "A game-theoretical approach for user allocation in edge computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 515–529, 2020.

[10] S. Saurav *et al.*, "Computational load balancing on the edge in absence of cloud and fog," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1499–1512, 2019.

[11] F. Bari *et al.*, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Serv. Manag.*, vol. 13, no. 4, pp. 725–739, 2016.

[12] J. Liu *et al.*, "Improve service chaining performance with optimized middlebox placement," *IEEE Trans. Netw. Serv. Manag.*, vol. 10, no. 4, pp. 560–573, 2017.

[13] I. Leyva-Mayorga *et al.*, "Inter-plane inter-satellite connectivity in dense LEO constellations," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3430–3443, 2021.

[14] Z. Qu *et al.*, "LEO satellite constellation for internet of things," *IEEE Access*, vol. 5, pp. 18391–18401, 2017.

[15] W. Rankothge *et al.*, "Data modelling for the evaluation of virtualized network functions resource allocation algorithms," *ArXiv*, vol. abs/1702.00369, Feb. 2017.