Switching-Aware Dynamic Control Path Planning for Software Defined Large-Scale LEO Satellite Networks with GEO Controllers

Tingting Zhang[1,3](✉), Fan Bai[2,3], Tao Dong[1,3], Jie Yin[1,3], Zhihui Liu[1,3],
and Yuwei Su[1,3]

[1] State Key Laboratory of Space-Ground Integrated Information Technology, Beijing Institute of Satellite Information Engineering, Beijing, China
zhangtingting@spacestar.com.cn
[2] Beijing Institute of Spacecraft System Engineering, Beijing, China
[3] China Academy of Space Technology, Beijing, China

Abstract. Recently, to acquire the programmability, flexibility and reconfigurability of network, the technologies of software-defined networking (SDN) are utilized to design new architectures for LEO satellite networks, where control plane is realized by several GEO satellites. But in the previous work, it is assumed that each LEO satellites could directly connect with the GEO controllers. This is unreasonable when the LEO data plane is a large-scale LEO satellite constellation, as the number of GEO controllers and antennas of each GEO controller are limited. In this work, we propose a design of software defined large-scale LEO satellite networks with limited GEO controllers, where the communication between GEO control plane and LEO data plane is achieved by limited cross-layer links. In this model, the selection of cross-layer links directly affects the load balance among GEO controllers and the latencies of control paths, which will deeply influence the latency of routing response. Thus, we propose a switching-aware dynamic control path planning scheme to handle this, where we propose a switching-time selection scheme to handle the impact of switching and a multi-objective optimization problem to deal with the selection of cross-layer links, and finally present a particle swarm optimization based algorithm to solve it. Simulation results demonstrate the better performance on reducing the bad influence of switching, optimizing the load balance among controllers and the average latencies of control paths.

Keywords: LEO satellite network · Software-defined networking · Multi-objective optimization component · Particle swarm optimization

1 Introduction

As the fast development of spaceflight technology, the ability of data processing of satellites has been greatly improved. Thus, recently, more and more research communities and industries are considering to achieve a global network service through satellites,

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 357, pp. 153–167, 2021.
https://doi.org/10.1007/978-3-030-69069-4_13

which are called satellite networks. During a satellite network, each satellite is set as a forward node or control node, and connects with other satellites by inter-satellite links (ISLs). Moreover, the satellite backbone network is expected to be designed as a LEO satellite constellation, because of its low-delay and real-time communication. Specially, a series of LEO satellite projects has been launched by some satellite entrepreneurs recently, such as OneWeb, SpaceX [1], which could herald the coming of large-scale LEO satellite networks.

Besides, to acquire the programmability, flexibility and re-configurability of network, the technologies of software-defined networking (SDN) are utilized to design new architectures for satellite networks [1, 2, 4, 5]. In the SDN-based satellite networks, the logically centralized control plane are usually realized by several controllers, in order to avoid the single point of failure and achieve better manageability of the network. Among the different types of satellites, the GEO satellite layer, with stationary position to the ground, wide coverage area, high communication capability and broadcast communication, is regarded as an appropriate layer to deploy the distributed control plane [6].

However, the previous work assumed that each LEO satellite could directly connect with the GEO controllers [6], which is unreasonable in the large-scale LEO satellite network. As the number of GEO controllers and antennas of each GEO controller are limited. For example, the satellite constellation proposed by SpaceX will contain more than 4425 LEO satellites. Suppose that each GEO satellite can hold n antennas pointing to LEO satellites, then it needs $4425/n$ GEO satellites to build direct connection with all LEO satellites. Even if $n = 10$, the number of GEO satellites is about 442, which is a much difficult task in practice. Even though the communication between LEO satellites and GEO controllers is realized by a broadcast control channel, it is also difficult for GEO controllers to provide enough power in order to broadcast to a huge number of LEO satellites. Thus, a practical solution is to select some special LEO satellites to connect with the GEO satellites, and all other LEO satellites communicate with GEO satellites through these special LEO satellites. In this scenario, links between LEO and GEO satellites are called cross-layer links, which will be frequently interrupted and connected by the relative motion of LEO and GEO satellites. Thereby, we need to plan proper cross-layer links before the current cross-layer links are interrupted, and compute control paths for each LEO satellites, which is called switching-aware dynamic control path planning (SADCPP) problem.

In this work, we propose a design of software defined large-scale LEO satellite networks with limited GEO controllers (SDLLSN), where the communication between GEO control plane and LEO data plane is achieved by limited cross-layer links. To handle the SADCPP problem, we propose a switching-aware dynamic control path planning scheme, where a switching-time selection scheme is proposed to handle the impact of switching, a multi-objective optimization problem is presented to deal with the selection of cross-layer links in each time slot. The problem of selecting cross-layer links (SCRLP) with considering the load balance among GEO controllers and the latencies of control paths is NP-hard. Finally we present a particle swarm optimization based algorithm to solve it.

2 Related Work

Software defined satellite networks have been deeply studied in [1, 2, 4–7]. Bao et al. [2] proposed an architecture of software-defined satellite networks, where the logically centralized entity is realized by GEO group. In the architecture presented by Xu et al. [6], the data plane is designed as a LEO satellite constellation, and the control of LEO date plane is realized through several GEO satellites, but it assumed the one-hop connection between GEO controllers and LEO satellites. And all these work are not involved the SADCPP problem.

Besides, there are also some work studying the dynamic topology of satellite network [8–11], but they do not consider the planning of cross-layer with considering the load balance among GEO controllers and the latencies of control paths. To the best of our knowledge, we are the first to study switching-aware dynamic control path planning problem in a large-scale LEO satellite network with limited GEO controllers.

3 System Model

In this section, we first give an architecture for software defined large-scale LEO satellite networks with limited GEO controllers (SDLLSN) and then elaborate the switching-aware dynamic control path planning problem.

3.1 An Architecture for Software Defined Large-Scale LEO Satellite Networks with Limited GEO Controllers

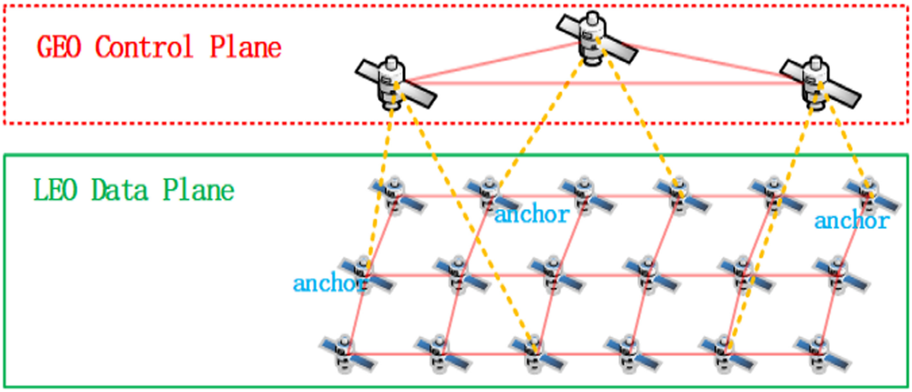

Fig. 1. The architecture of the SDLLSN

The proposed architecture of the SDLLSN is shown in Fig. 1. According to the researches mentioned above, the control plane layer consists of several GEO satellites, which have limited cross-layer links with LEO satellites. Besides, the control plane can be realized by a two-layer hierarchical controller architecture, where it will set each GEO controller just as a domain controller, but additionally deploy multiple super controllers in the ground station, as discussed in [6]. In such scenario, the GEO satellites must have at least one antennas pointing to the ground station such that they can communicate directly with their super controllers. The data plane layer contains an LEO satellite constellation, which can be assumed to be a Walker Delta Pattern constellation such that permanent inter satellite links can be constructed between LEO satellites. Finally, the communication between GEO control plane and LEO data plane is achieved by cross-layer links between the GEO and several LEO satellites, where these LEO satellites are called anchors.

3.2 Problem Formulation

First, note that a control path contains two segments, the direct path from LEO anchors to its GEO controller and the path from any LEO satellite to an anchor. Due to the relative motion between GEO and LEO satellites, the cross-layer links will be interrupted when the GEO and LEO satellites are not accessed. Therefore, antennas of the GEO satellites corresponding to these interrupting links must be re-planned to construct cross-layer links with other LEO satellites. The existing planning strategy make the GEO satellites select the LEO satellite nearest to it. This may result in long control path delays for some LEO satellites to its anchor. In SDLLSN, such long control path delays imply long latencies of responses to routing requests from data plane. Furthermore, the exact time that each cross-layer link is re-planned must also be handled carefully. As if too many cross-layer links are interrupted at the very same time point, it will seriously increase the path delays from LEO satellites to its controller, and thus will aggravate the response delays. Finally, the load balance among controllers is another important problem that has to be considered during the selection of anchors. Therefore, the dynamic control path planning problem should incorporate all these requirements. In this work, we design a switching-aware dynamic control path planning (SADCPP) scheme to handle it.

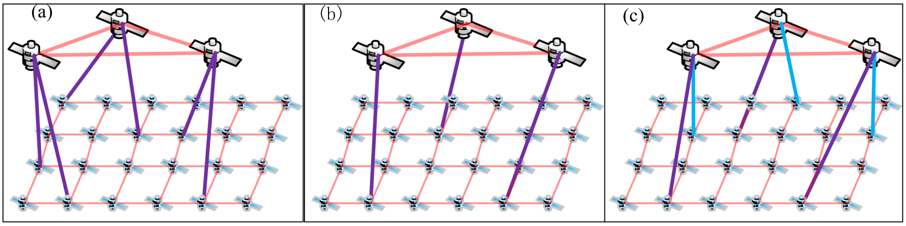

Fig. 2. Overview of switching-aware dynamic control path planning scheme

Now, we start to present how our SADCPP scheme work. In fact, SADCPP scheme consists of two mechanisms, the **TimeSelection** and the **AnchorSelection**. More precisely, the scheme will be run on one of the GEO controllers, which will be deployed these two mechanisms and perform as described in Fig. 2. The **TimeSelection** modular will be activated when the controller starts to re-plan the cross-layer links. Then the **TimeSelection** modular first collects the current state of all cross-layer links from other GEO controllers, and outputs the antennas that will be switched and the time when they will be switched. Then, the **AnchorSelection** modular will be activated to decide an anchor for each switching antenna, distribute controllers for all LEO satellites and compute the control path for each LEO satellite in the next time slot. Besides, the routing computation modular is responsible for planning new control path for LEO satellites that will lose its anchor during the switching. The controller will send all these policies to other controllers and the data plane. Then when the time goes to the point that the switching should be started, each switched antenna will stop the current connection and try to build its next connection depending on the policies it has received. And each LEO satellite will send its routing request through this new control path in the next time slot.

Next, before present details of the **TimeSelection** and **AnchorSelection** shemes, we first formalize the SADCPP problem. At first, we model the SDLLSN as a graph $G(t) = \langle V, E(t)\rangle$, where V consists of the GEO satellites V_c and LEO satellites V_s, and $E(t)$ is the link status in time t. In fact, $E(t) = E_c \cup E_s \cup E_{cs}(t)$, where E_c is the set of ISLs between GEO controllers, E_s is the set of ISLs between LEO satellites that keep permanent, and $E_{cs}(t)$ is the cross-layer links between the GEO control plane and LEO data plane. Note that for the LEO data plane, we only consider the permanent ISLs, because we want to avoid the influence of link switching in data plane on the transmission of control message. Furthermore, as the data plane is a LEO satellite constellation where permanent ISLs can be constructed between most of LEO satellites, this assumption is reasonable. Besides, in the following section, we will use $G_c = \langle V_c, E_c\rangle$ to represent the topology of the GEO control plane, $G_s = \langle V_s, E_s\rangle$ denote that of the LEO data plane with permanent ISLs, and set $G(t) = G_s \cup G_c \cup E_{cs}(t)$. The detailed notations and definitions used in this paper are summarized in Table 1.

Table 1. Notations and Definitions

Notations	Definitions	Notations	Definitions
$G(t)$	The topology of network in time t	m	The number of GEO controllers
G_c	The topology of GEO control plane	n	The number of nodes in V_s
V_c	Set of GEO controllers	M	The total number of antennas of all GEO controllers
E_c	Set of ISLs between GEO controllers	c	A controller in V_c
G_s	The topology of LEO data plane without dynamic ISLs	a	An antenna of a GEO satellite
V_s	Set of LEO satellites	S_{anchor}	Set of anchors
E_s	Set of permanent ISLs in LEO data plane	D_{as}	Access data of all GEO satellite with all LEO satellite

For the load balance among controllers, we represent it by the number of LEO switches controlled by each GEO controller. And the measure of load balance uses max–min fairness, which is defined as following:

$$\mathrm{FI} = \frac{\min\{num_i\}}{\max\{num_i\}}, i = 1, \ldots, m \tag{1}$$

where num_i is the number of LEO satellites controlled by the i-th GEO controller.

The control path delay is evaluated by average forward times from LEO satellites to their controller. Let L_i^s denote the forward times from i-th LEO satellites to an anchor s. Then the forward times from the i-the LEO satellite to its controller is defined as following:

$$L_i = min_{s \in S_{anchor}} \{L_i^s\} + 1 \tag{2}$$

where S_{anchor} is the set of anchors in LEO data plane.

Then, the problem can be formulated as given $G(t) = G_s \cup G_c \cup E_{cs}(t)$ in time t, we need to give an optimal decision of the following two things:

a. Based on the end time of each link in E_{cs}, decide the time when to switch such that for each GEO controller i, there is at least α_i cross-layer links still working during the switching, which is to guarantee the continuous communication between the GEO control plane and the LEO control plane.
b. The selection of anchors A for the switching antennas to connect in the next time slot such that the average latency of control paths is minimized, and the load balancing index FI is maximized. That is, our optimization objective is,

$$\min L_{ave} = \min \frac{1}{n}\left(\sum_{i=1}^{n} L_i\right) \text{ and } \max \mathrm{FI} \tag{3}$$

Subject to:

$$\forall i \in V_c \sum_{j \in V_s} e_{ij} \le \alpha_i \tag{4}$$

$$\forall j \in V_s \sum_{i \in V_c} e_{ij} \le \beta_j \tag{5}$$

$$\forall i \in V_c \sum_{j \in V_s} e_{ij} \ge \gamma_i \tag{6}$$

$$\forall j \in V_s L_j \le \delta \tag{7}$$

$$\forall e \in E_{switching} T_e \ge \sigma \tag{8}$$

$$T_{switching} = \Delta \tag{9}$$

In (4), α_i is the number of antennas in the i-th GEO controller, and then this equation limits the number of cross-layer links built from the i-th GEO controller. Similarly, Eq. (5) means that the cross-layer links built from the j-th LEO satellites is limited by the number of antennas β_j on it. Equation (6) requires that for every time slot, even during the switching, for the cross-layer links built from the i-th GEO controller, at least γ_i links keep working. Equation (7) means that the forward times of each LEO satellites is limited by δ. In Eq. (8), $E_{switching}$ is the set of cross-layer links that are waiting to be selected for the next time slot, T_e is the continuous time of cross-layer link e after the switching, and then this equation is to make sure that each new cross-layer link can work for an enough long time, that is, at least σ. Equation (9) defines the time needed to process the switching, and this implies that new cross-layer links will start to work in time $t + \Delta$, where t is the end time of this slot.

Then, for the selection of time to switch the cross-link, i.e., the **TimeSelection** mechanism, we will synthesize the end time of all antennas and the constraint of γ_i, which is processed as following:

- First compute the end time of each cross-layer links based on the orbits of GEO and LEO satellites.
- Then arrange all the antennas according to the end time in ascending order. Let $V_{switching}$ be the set of antennas waiting to switch, and $V_{switching}$ is empty at the start of our scheme. Then the sorted time series be defined as $t_1^{a_1}, t_2^{a_2}, \ldots, t_{\mathrm{M}}^{a_M}$, where $t_i^{a_i}$ represents the end time of antennas a_i, and $M = \sum_{i=1}^{m} \alpha_i$ is the number of antennas in all GEO controllers, i.e., the cross-layer links built between the control plane and data plane.
- Set $t_{end} = t_1^{a_1}$, and add a_1 to the set $V_{switching}$.
- For $i = 2, \ldots, M$, do as following:
- If $t_i^{a_i} - t_{end} \geq \Delta'$, break and go to the next step;
- Else, let c be the GEO satellite corresponding to a_i, then if the number of antennas of c in $V_{switching}$ has exceeded γ_c, break and go to the next step; else, add a_i to the set $V_{switching}$.
- Finally, output $V_{switching}$ and $t_{end} = t_{end} - \Delta$.
- For the scheme of setting $V_{switching}$, the reason behind it is that we want the antennas with close end time to be switched at the same moment. This will decrease the frequency of switching, and thus relieve the frequent updating of data plane caused by frequent switching cross-layer links. Thus, the cross-links stopping before $t_{end} + \Delta'$ will be switched at the same moment with the antenna a_1, where the parameter Δ' is used to control the time frequency of switching.
- Next, the selection of appropriate anchors A, i.e., the **AnchorSelection** mechanism, is more complicated. Let $E^i_{switching}$ be the set of LEO satellites that the antenna i can connect with, and N_i be its size. If the size of $V_{switching}$ (denoted by $\mathbb{M}$) is very small, we can use the brute force strategy, that is, for all the combination of switching cross-layer links, respectively compute $\frac{1}{n}(\sum_{i=1}^{n} L_i)$ and FI, and finally select the combination that result in optimal average latency of control paths and load balance index. However, when $\mathbb{M}$ is a little bigger, the complexity of this brute force strategy will become much higher. For example, for a 1584-satellite constellation, when we set $\sigma = 20$ min, N_i is more than 600. Then the complexity of such brute force strategy is $\prod_{i=1}^{\mathbb{M}} N_i$, which is more than $2^{9\mathbb{M}}$. Thus the brute force strategy is not suitable when $\mathbb{M}$ is a little big. Furthermore, note that the problem is a NP-hard combinatorial optimization problem.

Therefore, we propose an optimization algorithm to solve it, called **AS-PSO**, which is presented in the next section.

4 Discrete Particle Swarm Optimization Based Anchor Selection Algorithm

The particle swarm optimization (PSO) [12] is an intelligent, iterative optimization algorithm, which has the advantages of rapid convergence towards an optimum, fast and easy to compute. In this section, we will employ the discrete PSO version to solve the anchor selection problem in SDLLSN.

Let $V_{switching}$ be the set of waiting switching antennas generated by **TimeSelection** mechanism, and use $\mathbb{M}$ denote its size. Let $node_i$ is the set of LEO satellites that can access with the GEO controller corresponding to $a_i \in V_{switching}$ in the next time slot, and satisfies the requirement of Eq. (8). The size of $node_i$ is denoted by $\ltimes_i$. Then, the particle is represented by a $\mathbb{M}$ dimensional vector $\overrightarrow{p} = (p_1, p_2, \ldots, p_{\mathbb{M}})$, where p_i is an integer in $[1, \ltimes_i]$ and then $node_i(p_i)$ is the LEO satellite planned to connect with antennaa_i. Let $\bar{V}_{switching}$ denote the antennas that are not switched in the next time slot. Then by merging $\bar{V}_{switching}$ with$\{node_1(p_1), \ldots, node_{\mathbb{M}}(p_{\mathbb{M}})\}$, we obtain a possible anchor combination S_{anchor}. But to decide whether it is a feasible solution, we still need to check the following conditions:

- For each anchor s in S_{anchor}, check if s satisfies the requirement of Eq. (5), i.e., $\sum_{i\in V_c} e_{is} \leq \beta_s$;
- For each LEO satellite i in V_s, check if $(min_{s\in S_{anchor}} \left\{L_i^s\right\} + 1) \leq \delta$, where L_i^s is computed by using the shortest path algorithm Dijkstra.

If both of the above two conditions are satisfied, the anchor combination induced by this particle is a feasible solution.

The particle's velocity, local best position, global position and their updating strategies are defined as common discrete- PSO. Besides, we also employ the adaptive parameters approach based on [12] to avoid premature convergence.

Furthermore, for the evaluation of fitness, we still need to select a controller for each LEO satellite, where we propose a load balancing aware clustering (LBAC) scheme to handle it. Then, based on the output of LBAC, we can compute the load balance index FI and the average forward times L_{ave}. Only if both FI and L_{ave} are better, we say that the fitness of this particle is better. Now, we have overview the algorithm AS-PSO, and details of the algorithm are presented in Algorithm 1.

Algorithm 1. Discrete Particle Swarm Optimization Based Anchor Selection Algorithm (AS-PSO)

```
Input: $G(t), V_{switching}, t_{end}, D_{as}, \mathbb{T}, \mathbb{p}$
Output: $S_{anchor}^{op}$
for each antenna $a_i^c \in V_{switching}$, based on $D_{as}$, compute $node_i$ such that $\forall s \in node_i, T_{e(c,s)} \geq \sigma$ where $e(c,s)$ is the cross-layer link between LEO node $s$ and GEO controller $c$.
iteratively generate $\mathbb{p}$ particles $\overrightarrow{p_1}, \ldots, \overrightarrow{p_{\mathbb{p}}}$ randomly such that each particle corresponds to a feasible solution, and then generate the initial velocity $\overrightarrow{v_1}, \ldots, \overrightarrow{v_{\mathbb{p}}}$, local best position $\overrightarrow{p_1}', \ldots, \overrightarrow{p_{\mathbb{p}}}'$.
for the solution corresponding to each particle, employ the LBAC algorithm to compute the load balance index and average forward times, and record the best values $\mathrm{FI}^{op}$, $L_{ave}^{op}$, and the global best particle $\overrightarrow{p_{best}}$.
for $it = 1, \ldots, \mathbb{T}$ do
  update the inertia weight.
  for $\iota = 1, \ldots, \mathbb{p}$ do
    update the velocity $\overrightarrow{v_\iota}$ and particle $\overrightarrow{p_\iota}$ by employing the common discrete PSO stratgy.
    if $\overrightarrow{p_\iota}$ corresponds to a feasible solution, then compute its fitness values $\mathrm{FI}^\iota$, $L_{ave}^\iota$. And then update $\mathrm{FI}^{op}$, $L_{ave}^{op}$, $\overrightarrow{p_\iota}'$ and $\overrightarrow{p_{best}}$, if these fitness values are better.
    if $\overrightarrow{p_\iota}$ does not correspond to a feasible solution, replace $\overrightarrow{p_\iota}$ with a randomly generated particle, and then check and update $\mathrm{FI}^{op}$, $L_{ave}^{op}$, $\overrightarrow{p_\iota}'$ and $\overrightarrow{p_{best}}$ as above.
  endfor
endfor
```

Finally, for a selection of anchors, we present the LBAC scheme to decide a controller for each LEO satellite .Then for each LEO satellite The LBAC scheme first computes shortest paths from each LEO satellite to all GEO controllers. i, the LBAC counts the control paths whose delay is no more than other control paths, and then add the GEO controllers corresponding to these control paths to the controller set C_i of i-th LEO satellite. If C_i is equal to1, directly set this unique element as the controller of this i-th LEO satellite. After the iteration is completed, iterative all the LEO satellite again. During each iteration$i$, if C_i is greater than1, set the controller of the i-th LEO satellite to be the element in C_i that has controlled the minimal number of LEO satellites. Finally, according to the above clustering result, compute the load balancing index FI and the average latency of control pathsL_{ave}.

Algorithm 2. Load Balancing-Aware Clustering (LBAC) Algorithm

```
Input: a selection of anchors $A_1, \dots, A_m$ respectively for each GEO controllers
Output: FI and $L_{ave}$
initialize a series of empty sets $C_1, \dots, C_n$, and parameters $num_1, \dots, num_m$ to $0$.
for $i = 1, \dots, n$ do
    employ the Dijkstra algorithm to compute the shortest path from the $i$-th LEO satellite to each GEO controller, and computes the shortest path delay as $L_i^1, \dots, L_i^m$.
    let $L_i^{min} = \min\{L_i^1, \dots, L_i^m\}$, iteratively check $L_i^1, \dots, L_i^m$, if $L_i^j$ is equal to $L_i^{min}$, add $j$ to $C_i$.
    if the size of $C_i$ is equal to 1, then set the controller of this $i$-th LEO satellite as the unique element of $C_i$. Suppose the unique controller is $in$, then update $num_{in}$ as $num_{in} + 1$.
endfor
for $i = 1, \dots, n$ do
    if the size of $C_i$ is greater than 1 do
    iteratively check each controller in $C_i$, and let $j$ is the index of controller such that $num_j$ is minimal.
        Then set $j$ as the controller of this $i$-th LEO satellite
    endif
endfor
finally, compute FI and $L_{ave}$ according to the above clustering result.
```

5 Numerical Simulation and Analysis

In this section, we will present the detail simulation scenario, and then we analyze the numerical simulation results of our SA-DCPP scheme.

5.1 Simulation Setup

In our simulation scenario, the LEO data plane is Walker Delta Pattern constellation introduced by the Starlink project. The parameters comes from the proposal submitted by SpaceX to FCC in 2018, which is presented in the Table 2. In such satellite constellation, each LEO satellite can build permanent ISLs with LEO satellites in its neighbor orbits. Then, for the network topology, each LEO satellite can hold permanent four ISLs with the two satellites in the same orbit and the two satellites in its two neighbor orbits, and one cross-layer ISL with some GEO satellite.

For the GEO control plane, we design six GEO satellites with altitude 35,786 km, and each of them has four antennas pointing to the LEO data plane. Table 3 describes the parameters of them.

To simulate this constellation, we obtain the two-line element sets of the 60 satellites launched by SpaceX, of which the NORAD numbers are from 44235 to 44294U [13]. Then, we analyze the right ascension of the ascending node (RAAN) of the J2000 coordinate system. Finally, the epoch of the seed satellite is set as 27 May 2019 06:14:00.000

Table 2 The parameters of LEO satellite constellation

Parameters	Value	Parameters	Value
Orbital Planes	24	Altitude	550 km
Satellites Per Planes	66	ISLs Per LEO Satellite in LEO Data Plane	2 inter-ISLs, 2 intra-ISLs
Total Satellites	1584	Cross-layer ISLs per LEO Satellite	1
Inclination	53°		

Table 3 The parameters of LEO satellite constellation

Parameters	Value	Parameters	Value
Inclination	0°	ISLs Per GEO satellite in the Control Plane	2
Number of GEO Satellites	6	Cross-layer ISLs per GEO Satellite	4
Altitude	35786 km	Total Cross-layer ISLs	24

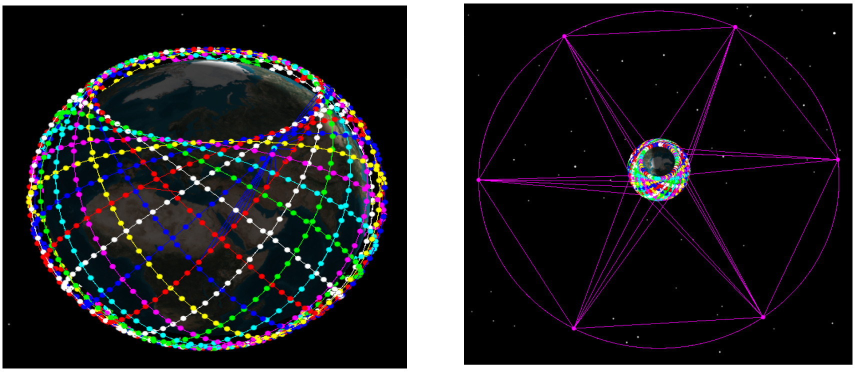
(a) (b)

Fig. 3 (a) Simulation diagram of LEO satellite constellation and (b) Simulation diagram of cross-layer ISLs

UTC, the RAAN is 160°, the eccentricity is 0°, and other parameters are 0. Then, we can generate the LEO satellite constellation, as shown in Fig. 3(a).

Then, the GEO satellites is simulated, and we can get the time-variant data of the access between GEO and LEO satellites, and the latency of ISLs and cross-layer links. Figure 3(b) shows the cross-layer ISLs at some point. The simulation shows that the number of LEO satellites accessed by a GEO satellite in a time duration 20 min is more than 600, the latency of cross-layer ISLs varies from 117 to 148 ms, and the latency of ISLs between LEO satellites varies from 3.7 ms to 6.1 ms. This implies that if two different ISLs have same forward times, the difference of the path delay between them is small. Thus, we evaluate the optimization objective L_{ave} by the forward times of control paths, instead of the path delay, for which our optimization will be immune to the influence of small difference of control paths.

We simulated the state of our satellite network during 24 h, and recorded the time variant data. Based on these data, we then simulate our SA-DCPP scheme by using MATLAB. The simulation results are presented in the next section.

5.2 Performance Evaluation

Now we analyze the performance of the algorithms on switching-aware, load balance among controllers and control path latencies as follows.

Figure 4 shows the switching of the cross-layer links connected with one GEO satellite during 120 min. In fact, we have simulate for 24 h, and finally generate more than 200 time slots excluding the switching duration. But to clearly demonstrate the results, we only present the results about one GEO satellite during 120 min. The Y-axis is the index of each LEO satellite, and the X-axis is the time from 100 to 220 min. From Fig. 4, we can see that during each switching, every GEO controller still has at least one cross-layer link connected with the LEO data plane, and each cross-layer link can keep working more than 15 min.

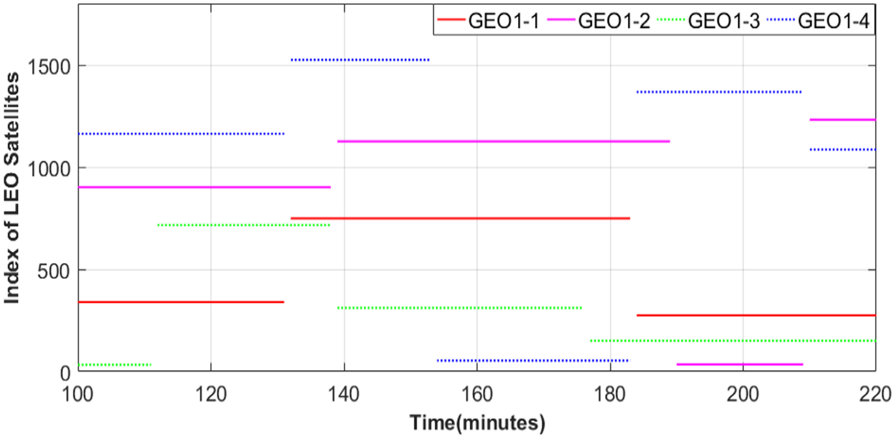

Fig. 4 Switching of the Cross-layer Links

Figure 5 shows the load balance among GEO controllers. The Y-axis is the number of LEO satellites, and the X-axis is the time slot from 0 to 10. And similarly to have a good demonstration, we only Fig. 10 time slots (excluding the switching duration). From Fig. 5, it is obvious that our algorithm can obtain a good load balance, and the worst result is at the first time slot which is influenced by the bad initial selection of anchors. And for most time, the algorithm can achieve best load balance, as shown in Fig. 6.

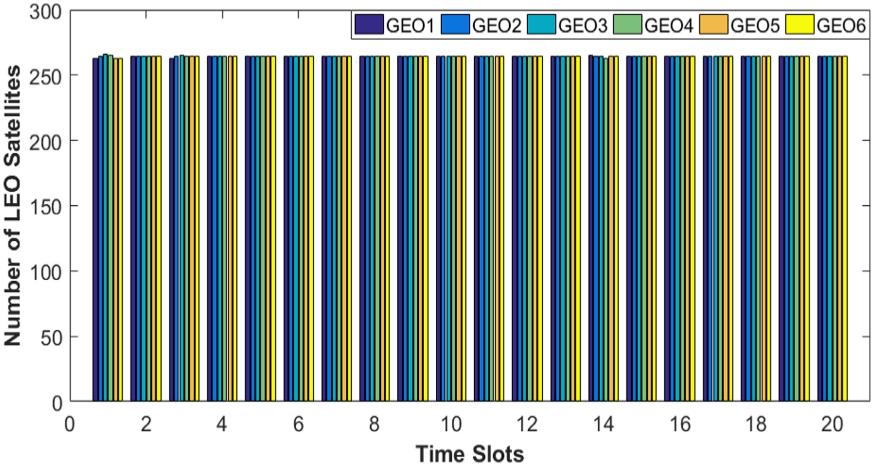

Fig. 5 Load Balance among GEO Controllers

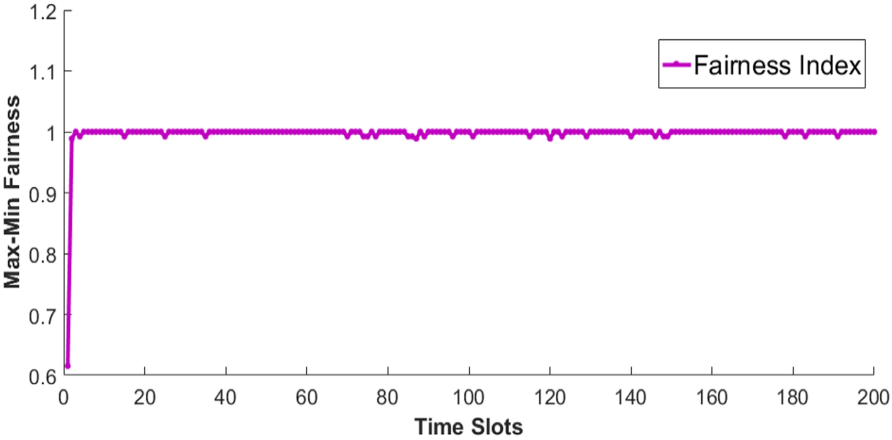

Fig. 6 Max–Min Fairness Index

Figure 7 shows the average and maximum forward times of control path during the 200 time slots (excluding the switching duration). At the first time slot, the maximum forward times from LEO satellites to its controller is 20, and the average forward times is about 7. But after employing our SADCPP scheme, the maximum forward times is decreased to 11 and the average forward times is reduced to about 5. Moreover, to evaluate the control path delay, we further compute the average and maximum control path delays, which is presented in Fig. 8. The propagation delay of ISLs and cross-layer links is evaluated. The results show that after employing our SADCPP scheme, the average control path delay can be reduced to about 155 ms (after the 80-th time slot), and the maximum control path delay is about 186 ms. The long latencies are mainly caused by the long propagation delay of cross-layer links, and our algorithm can actually decrease the forward times resulted from the large number of nodes in LEO data plane.

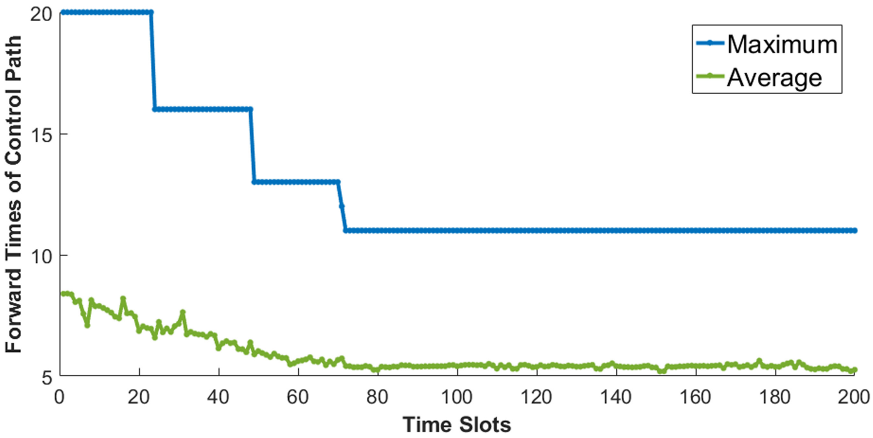

Fig. 7 Average and Maximum Control Forward Times

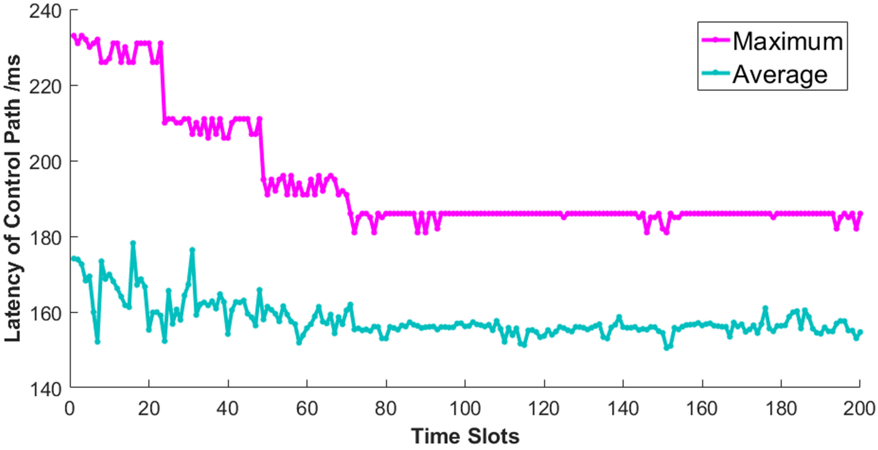

Fig. 8 Average and Maximum Control Path Delay

6 Conclusion

In this work, we propose a practical controller architecture for the software defined large-scale LEO satellite networks, where with considering the limited number and antennas of GEO satellites, we assume that the communication between GEO control plane and LEO data plane is achieved by limited cross-layer links. Then, to plan proper cross-layer links, we propose a switching-aware dynamic control path planning (SADCPP) scheme, consisting of the TimeSelection and the AnchorSelection mechanisms. The TimeSelection can decide the next switching time with maintaining enough cross-layer links in the switching duration and building new cross-layer links that last long enough. Then the AnchorSelection scheme compute proper anchors by employing a heuristic algorithm AS-PSO, which is realized to solve the cross-layer link selection problem with considering the load balance among GEO controllers and the latencies of control paths. Simulation results demonstrate the better performance of our SADCPP scheme on reducing the bad influence of switching, optimizing the load balance among controllers and the control path delays of LEO satellites.

References

1. Majumdar, A.K.: Optical Wireless Communications for Broadband Global Internet Connectivity: Fundamentals and Potential Applications. Elsevier (2018)
2. Bao, J., Zhao, B., Yu, W., et al.: OpenSAN: a software-defined satellite network architecture. ACM SIGCOMM Computer Communication Review. ACM, **44**(4), 347–348 (2014)
3. Tang, Z., et al.: Software defined satellite networks: benefits and challenges. In: IEEE Computers, Communications and IT Applications Conference, pp. 127–132. IEEE (2014)
4. Gardikis, G., et al.: NFV applicability and use cases in satellite networks. In: European Conference on Networks and Communications (EuCNC), pp. 47–51. IEEE (2016)
5. Li, T., Zhou, H., Luo, H., et al.: SERvICE: a software defined framework for integrated space-terrestrial satellite communication. IEEE Trans. Mob. Comput. **17**(3), 703–716 (2017)
6. Xu, S., Wang, X.W., Huang, M.: Software-defined next-generation satellite networks: architecture, challenges, and solutions. IEEE Access **6**, 4027–4041 (2018)
7. Barritt, B.J., Eddy, W.: SDN enhancements for LEO satellite networks. In: 34th AIAA International Communications Satellite Systems Conference (2016). https://doi.org/10.2514/6.2016-5755
8. Li, Y., Wang, Y., Zhang, Q., Yang, Z.: TCDS: a time-relevant graph based topology control in triple-layer satellite networks. In: IEEE Wireless Communications Letters (2019)
9. Bouttier, E., et al.: Analysis of content size based routing schemes in hybrid satellite / terrestrial networks. In: IEEE Global Communications Conference (GLOBECOM), pp. 1-6. IEEE (2016)
10. Sara El Alaoui, Byrav Ramamurthy: Routing Optimization for DTN-Based Space Networks using a Temporal Graph Model. ICC (2016). https://doi.org/10.1109/ICC.2016.7510733
11. Kondrateva, O., et al.: Throughput-optimal joint routing and scheduling for low-earth-orbit satellite networks. In: 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS), pp. 59–66. IEEE (2018)
12. Gang, X.: An adaptive parameter tuning of particle swarm optimization algorithm. Appl. Math. Comput. **219**(9), 4560–4569 (2013)
13. https://www.space-track.org/