Achieving Scalable and Efficient Routing in LEO Constellations

Kejun Li[1], Changlin Jiang *[1], Yu Wang[1], Dan Zhao[1], Qing Li[1], and Yong Jiang[1,2]

[1]Department of Strategic and Advanced Interdisciplinary Research, Peng Cheng Laboratory, Shenzhen, China
[2]Tsinghua Shenzhen International Graduate School, Shenzhen, China

***Abstract*—Mega Walker Delta constellations, such as Starlink and OneWeb, can offer economic seamless ubiquitous coverage solutions. However, its highly topological dynamics bring about serious challenges to network routing. Existing solutions either suffer from continuous convergences with poor scalability, or do not take into account the effects of the polar region on the Inter-Satellite Links. In this paper, we explore the unique characteristics of Walker Delta constellation, and present SER, a scalable and efficient routing algorithm based on the topology features. SER only uses orbit numbers and satellite numbers of source and destination to calculate the shortest path, therefore, flooding link state information is not required. SER has $O(1)$ computational complexity. Furthermore, SER only needs to be executed once at the source node, who writes the routing decision into packets, and subsequent nodes along the path adopt source routing to forward packets. Evaluation results show that the execution time of SER is always kept at about $40us$, regardless of the network scale, and 94.03% of the paths output by SER are no more than 10% longer than those output by OSPF.**

***Index Terms*—Walker Delta constellation, Inter-Satellite Link, Routing**

I. Introduction

In recent years, the latest advances in satellite manufacturing and launch technologies have nurtured the booming of Low Earth Orbit (LEO) satellite constellations, especially mega Walker Delta constellations such as Starlink [1] and OneWeb [2]. LEO satellite constellations can offer economic and seamless ubiquitous coverage solutions for areas where it is difficult to establish base stations, such as deserts, oceans, and sparsely populated remote areas. Moreover, LEO satellite constellations are also ideal for special scenarios such as disaster relief and emergency communications, when terrestrial communication facilities are destroyed.

Despite the aforementioned advantages, integrating the satellite networks and the terrestrial networks for global Space-Terrestrial Integrated Networks (STINs) faces various technical challenges, especially, the highly dynamic topologies of satellite networks bring about technical challenges to network routing. Zhang et al. found that for a LEO with 450 satellites, the topology change interval is less than 20 seconds [3]. The widely deployed routing protocols in terrestrial network, such as OSPF and BGP, are no longer suitable for mega satellite constellations, since the frequent route updates in response to topology changes would bring excessive pressure to the limited on-board computing and storage sources.

To mitigate the negative effects brought by topological dynamics, many dedicated routing schemes for satellite networks are proposed. Some schemes convert dynamic topologies to logical static topologies by dividing space area or terrestrial area into cells, and compute routes based on the address of cells rather than the satellites [4, 5, 6, 7]. However, these schemes either only apply to Walker Star constellations but not Walker Delta constellations, which are adopted by most mega satellite constellations, or do not take into account the effects of the polar region on the Inter-Satellite Links (ISLs). Some schemes optimize the conventional routing protocols to reduce the frequency of routing calculation caused by topology changes [8, 9]. However, with the continuous expansion of the network scale, such schemes still cannot guarantee the scalability. Thomas et al. proposed to calculate the shortest paths using only the orbit number and satellite number[10]. However, it only considers the minimum hops path between the source orbit and the target orbit, but excluding the hops within the target orbit. So the end-to-end paths found may not be the shortest ones.

In this paper, we propose a scalable and efficient routing scheme SER (Scalable and Efficient Routing) for Walker Delta constellation. First, we reveal change patterns of satellite numbers and orbit numbers along the shortest paths in Walker Delta constellation. We also observe the existence of High-Latitude-Loops, i.e., the closed loops around the polar regions that can be followed to bypass the polar regions and reach the target orbit. Based on these discoveries, SER adopts pole-crossing or continuously following HLL to address the negative impacts of the polar region on the ISLs, and calculates the end-to-end shortest path by enumerating all routing candidates. To mitigate the negative effects brought by topological dynamics and ensure routing scalability, SER only uses the orbit numbers and satellite numbers of the source satellite and target satellite, thus avoiding flooding of link state information within the topology. The computational complexity of SER's routing algorithm is $O(1)$. SER adopts source routing to further reduce the workload of intermediate nodes, which fits well with the resource-constrained and constantly-dynamic satellite network. We conduct a series of simulations to validate our design, and the results show that the execution time of SER is always kept at about $40us$, regardless of the network scale, and 94.03% of the paths output by SER are no more than 10% longer than those output by OSPF.

II. Background and Related Work

In this section, we first use OSPF as an example to analyze why conventional terrestrial routing protocols are not suitable for satellite networks, and then discuss related works.

A. Why OSPF is not suitable for mega satellite constellations

To evaluate the performance of OSPF in satellite network, We build a simulation platform based on NS3+DCE+QUGGA [11], and simulate a LEO satellite network containing 156 nodes (12 orbits, each containing 13 satellites, with a phase factor of 5). Hello interval and dead interval of OSPF are set to $10s$ and $40s$, respectively.

During an one-hour simulation, we observe that an average of 11 update events and a maximum of 28 update events occur per minute. It can be inferred that in a mega satellite constellation with a much larger network scale, the frequency of triggering update events will be much higher. Since each OSPF update event will trigger the synchronization of LSDB (Link State DataBase) and route calculation across the network, onboard capabilities will be overwhelmed, and the network may become unstable, even resulting in routing loop. Therefore, OSPF is not suitable for mega satellite constellations.

B. Related Work

Optimized conventional routing protocols. OSPF+ [8] uses topology predictions to determine the topological changes caused by polar region, and optimizes the synchronization mechanism of LSDB across the network by extending the neighbor state machine, resulting in more rapid route convergence. DT-TCA [9] also uses topology prediction to aggregate multiple route updates within a period into one update, thus reduces route update frequency and mitigates route jitter caused by network topology changes. However, with the increase of the scale of satellite network, satellites' on-board processing capabilities struggle to meet the demand required for predictions.

Location-based routing. DRA [5] uses the orbit number and satellite number as a satellite's virtual address to calculate the next hop based on the relative position information. It waives the exchanges of link state information between satellites. OPSPF [12] ignores topology changes caused by polar regions, so as to decrease the frequency of exchanging link state information, and reduce the route calculation cost caused by topology changes. However, these two solutions only apply to the walker star constellation and do not take into account the impacts of *seams [5]* on routing. Thomas et al. propose to calculate the shortest paths using only the orbit number and satellite number [10]. Liu et al. divide the earth surface into several regions, and calculate routing based on the region number of the source and the target [7]. However, these two schemes only consider hops between the source orbit and the target orbit when calculating the routes, and ignore the hops within the target orbit, so the end-to-end path may not be the shortest path.

III. Characteristics of Walker Delta Constellation

For ease of description, unless otherwise specified, the constellation in this paper refers to *Starlink phase 1 shell 4 group 4*. Since SpaceX has not officially disclosed the relevant parameters of the constellation, we set the key parameters of the constellation based on public literature [13] as follow: orbit number N is 72, the number of satellites in each orbit M is 18, the phase factor F [14] is 45, and an inclination θ [14] of 53.2°. We use $\langle p, s\rangle$ to identify a satellite, where $p \in [0, N-1]$ and $s \in [0, M-1]$ denote the orbit number and satellite number of the satellite, respectively. $(N/M/F/\theta)$ is used to denote a constellation.

We assume each satellite has four neighboring satellites: two in the same orbit, and the other two in the left and right orbits, respectively. We call the links between satellites in the same orbit and satellites in different orbits as intra-ISLs and inter-ISLs, respectively. The intra-ISLs are maintained all the times, while the inter-ISLs would be shut down in polar areas and reestablished outside of the polar areas [15].

Since $F > \frac{N}{2}$, we argue that the inter-ISLs connect two satellites from adjacent orbits whose satellite numbers differ by 1 would increase the link stability, compared with connecting two satellites with the same satellite numbers. Therefore, we present SER's design rational and details based on Starlink, and the inter-ISLs connect two satellites whose number differ by 1. Note that SER can also be applied to any other Walker Delta constellations and other inter-ISLs connecting principle.

A. Change patterns of the orbit numbers and satellite numbers

We observe following change patterns of the orbit numbers and satellite numbers of satellites connected by inter-ISLs.

Observation 1. *For two satellites connected by an inter-ISL, if the orbit numbers of the two satellites are not 0 and $N-1$, then their orbit numbers and satellite numbers satisfy: if the orbit number of one satellite is larger than the orbit number of the other satellite by 1, then its satellite number is smaller than that of the other satellite by 1, and vice versa. And if the orbit numbers of the two satellites connected by an inter-ISL are 0 and $N-1$, the absolute value of the difference between the satellite numbers of the two satellites is not 1 any more.*

Fig.1 illustrates an example of the above observation based on a constellation ($N = 72$). The blue inter-ISL connects satellites $\langle 0, 8\rangle$ and $\langle 1, 7\rangle$. It can be seen that the satellite number decreases from 8 in $0-th$ orbit to 7 in $1-st$ orbit. The red inter-ISL connects two satellites identified by $\langle 71, 0\rangle$ and $\langle 0, 8\rangle$. It can be found that the satellite number changes from 0 in 71-th orbit to 8 in 0-th orbit. The reason behind this number hopping is the accumulation of phase offset [15]. We further present the number hopping patterns between 0-th orbit and $(N-1)$-th orbit in the following lemma and theorem.

Lemma 1. *Starting from satellite $\langle p, s\rangle$, after traversing N continuous inter-orbit hops and getting back to source orbit again, i.e., the p-th orbit, the satellite number of the reached* *satellite in the p-th orbit is* $((s-(N-F)+c*M) \bmod M)$*, where c is a non-negative integer to ensure satellite number* $\in [0, M-1]$.

Theorem 1. *Assume an inter-ISL connecting two satellites from the two adjacent orbits* 0*-th and* $(N-1)$*-th orbit, the satellite numbers of these two satellites,* s_0 *and* s_{N-1}*, satisfy:*

$$s_0 = (s_{N-1} + F - 1) \bmod M, \quad (1)$$

$$s_{N-1} = (s_0 - F + 1 + c*M) \bmod M. \quad (2)$$

See ref. [16] for the proof of Lemma 1 and Theorem 1.

If the orbit number and satellite number of source satellite are known, the satellite number of the satellite on the reached orbit after a number of continuous inter-orbit hops can be derived by combining *Theorem 1* and *Observation 1*.

B. High-Latitude-Loop

Through 3D simulation, we find that there exist loops near the polar regions, which are called the High-Latitude-Loops (HLLs). Because the satellites in the Walker Delta constellation have two directions of motion [7, 10] (from south to north, and from north to south), there are four high-latitude loops at both poles.

Fig.2 shows part of a HLL around the north pole. The yellow lines represent inter-ISLs and the green lines are intra-ISLs. Assume that a packet needs to be sent from satellite $\langle 32, 1\rangle$ to satellite $\langle 22, 8\rangle$. Satellite $\langle 30, 3\rangle$ can be reached through the two inter-ISLs from $\langle 32, 1\rangle$, i.e., $\langle 32, 1\rangle \rightarrow \langle 31, 2\rangle \rightarrow \langle 30, 3\rangle$. Since $\langle 29, 4\rangle$ is already inside the polar region, the inter-ISL between $\langle 30, 3\rangle$ and $\langle 29, 4\rangle$ is broken (indicated by the red line). So the packet is first sent from $\langle 30, 3\rangle$ to $\langle 30, 2\rangle$ through an intra-ISL, and then reaches $\langle 27, 5\rangle$ through 3 inter-ISLs. The target satellite $\langle 22, 8\rangle$ will be eventually reached by continually following the HLL. The purple lines show the entire path.

Moreover, the orbit numbers and satellite numbers of satellites along a HLL follow the observation below.

Observation 2. *The orbit numbers in a HLL change monotonically. If the orbit numbers of of satellites connected by inter-ISLs increase along the routing path, the satellite numbers of satellites connected by intra-ISLs also increases, and vice versa.*

As Fig.2 shows, along the routing path shown by the purple arrows, the orbit numbers of satellites connected by inter-ISLs decrease monotonically in a HLL, i.e., $32 \rightarrow 31 \rightarrow \cdots \rightarrow 22$, while the satellite numbers of satellites connected by intra-ISLs also decrease. For example, for the intra-ISL $\langle 30, 3\rangle \rightarrow \langle 30, 2\rangle$, the satellite number decreases from 3 to 2.

Fig.2 also shows that intra-ISLs and inter-ISLs appear alternatively on a HLL and an intra-ISL is followed by a group of inter-ISLs, and the numbers of inter-ISLs in HLLs exhibit a "3/3/2" pattern, i.e., the sequence of "one intra-ISL + three inter-ISLs + one intra-ISL + three inter-ISLs + one intra-ISL + two inter-ISLs" repeats throughout the HLL.

IV. Design

A. Design Rational

Ignoring the effects of the polar areas on the connectivity of the inter-ISLs and assuming that the lengths of inter-ISLs and intra-ISLs are the same, the satellite network can be regarded as a Manhattan street network [17]. For each source-destination pair, there may exist more than one minimum-hops path. Along such paths, *the orbit number and satellite number increases or decreases monotonically* [17].

While routing in a Manhattan Street networks is well-established, routing in actual Walker Delta constellations faces two challenges. First, when latitude exceeds a threshold near the polar regions, the inter-ISLs are temporarily down. Second, the length of inter-ISL fluctuates as the latitude of the satellite changes, i.e., the higher the latitude is, the shorter the link length is. When designing the routing mechanism in Walker Delta constellations, it is necessary to consider:

- how to avoid the out-of-service inter-ISLs inside the polar regions;
- how to find the path with the shortest end-to-end length when the lengths of inter-ISLs vary across the network.

1) Impact of polar regions: According to the analysis in Section III-B, if a polar region is encountered along the path, the routing algorithm may either choose the pole-crossing option to cross the polar region or the HLL option to bypass the polar region. Since an end-to-end path may encounter the polar regions more than once, there are often many routing candidates.

2) Finding the shortest end-to-end path: The end-to-end path length (L) can be calculated as follows:

$$L = H_1 * L_{intra} + H_2 * L_{inter}, \quad (3)$$

where H_1 and H_2 represent the numbers of intra-ISLs and inter-ISLs, respectively, L_{intra} and L_{inter} are the length of intra-ISL and inter-ISL, respectively.

Assume that the earth is a standard sphere, L_{intra} can be derived as:

$$L_{intra} = 2R * \sin\left(\frac{\pi}{M}\right), \quad (4)$$

where *R* is the height of the orbit.

Since the length of inter-ISL varies with the latitude, to evaluate the influence of latitude on the length of end-to-end path, we take a "snapshot" of the network topology, which is considered to be static. Then, we choose satellite $\langle 0, 0\rangle$ as the source satellite, randomly select a satellite from each orbit as the destination satellite, and analyze the lengths of the minimum-hop paths for each source-destination pair. The analysis results show that the maximum length difference is less than 6%. Therefore, in order to simplify the design of the routing algorithm, we assume that *all the inter-ISLs have the same length.* According to Fig.3, We can approximately calculate L_{inter} as follows:

$$L_{inter} = 2R * \sin\left(\frac{\pi}{N}\right) * \cos\left(\frac{2\pi}{M*N} * (N-F)\right), \quad (5)$$

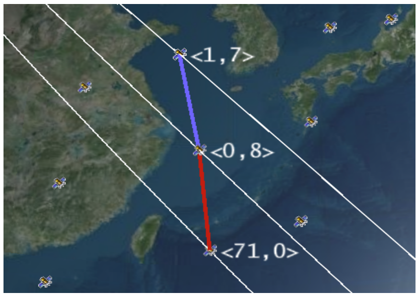

Fig. 1: Change pattern of the orbit numbers and satellite numbers.

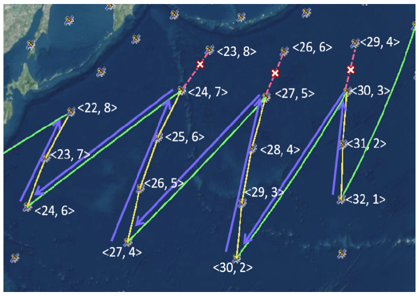

Fig. 2: An example of routing through HLL.

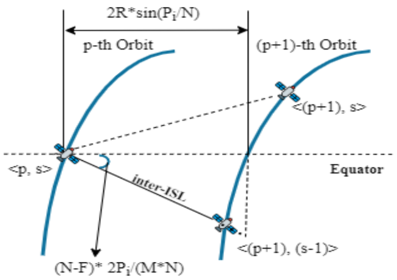

Fig. 3: Illustrative instructions for calculating length of inter-ISL.

The shortest path can be obtained by enumerating all the different routing candidates, calculating the length of path for each candidate, and choosing the shortest one.

B. Solution

In the proposed routing scheme, the source satellite determines the shortest path using Algorithm 1 and writes the path information into packets. Then, the intermediate nodes along the path perform source routing following the procedure described in Sec. IV-B2.

1) Shortest Path Calculation: To calculate the shortest path, two constants are used, namely H_{INTRA} and H_{INTER}. H_{INTRA} represents the number of intra-ISLs needed to cross the polar region, and H_{INTER} denotes the number of consecutive inter-ISLs between two adjacent polar regions. The values of these two constants are closely related to constellations' parameters, which can be obtained by analyzing the simulation data of constellations. For example, we simulate the constellation (72/18/45/53.2°) running for one period of revolution (about 94 minutes), and the latitude threshold is set to 45° N and 45° S. A snapshot of the constellation is taken every minute, OSPF calculates shortest paths for $1296 * 1295$ source-destination pairs based on each snapshot. By analyzing the shortest paths calculated by OSPF, we find that OSPF uses 3 hops (92.94%) or 4 (7.06%) hops to cross the polar region. Therefore, H_{INTRA} is set to 3. Through the same analysis method, the value of H_{INTER} is determined to be 16.

Since the earth is a sphere, there are two available routing directions for source satellite to reach destination satellite, one with monotonically increasing orbital number, and the other one with monotonically decreasing orbital number. In order to find the shortest path, the routing algorithm considers both routing directions (line 1), and calculates a shortest path for each routing direction. The shorter path is selected as the final shortest path.

For each routing direction, four variables are first calculated (line 2):

(1) ***D***: the number of inter-ISLs between source and destination, which equals the absolute difference between the orbit numbers of the source and destination.

(2) ***S'***: the satellite number in target orbit through D consecutive inter-ISLs without considering polar region, which can be obtained according to Observation 1&2.

Algorithm 1 Algorithm of calculating the shortest path

```
Require: $p_s, s_s, p_d, s_d, stp = +\infty$
Ensure: $PathParameters$                    ▷ info. of the shortest path
 1: for $A$ in (1, -1) do                    ▷ both routing directions
 2:     Calculates $D, S', H_P, T$
 3:     for $t = 0, T$ do                  ▷ Enumerating (T+1) routing
        candidates
 4:         Calculates $H_L, H_T$
 5:         if $stp > (D * L_{inter} + (H_C + H_L + H_T) * L_{intra})$
        then
 6:             PathParameters = $\{A, t\}$
 7:         end if
 8:     end for
 9: end for
10: return PathParameters
```

(3) ***H_P***: the nubmer of inter-ISLs between source satellite and the first polar region, which can be obtained based on the latitude of source satellite.

(4) ***T***: the times of encountering polar region along the path without considering HLL. $T = 1 + (D - H_p)/H_{INTER}$.

After calculating these variables, the algorithm starts to calculate the shortest path (line 3-8). According to Sec. IV-A1, polar region can either be crossed through H_{INTRA} intra-ISLs, or be bypassed by following HLL. It can be easily proved by contradiction that the sequence of the polar-crossing and following HLL along the path has no effect on the length of the path. Therefore, when calculating the path length, the polar-crossing is preferentially adopted. If the path encounters polar regions for T times, there are $(T + 1)$ routing candidates:

(1) Bypassing polar region through HLL until reaching the target orbit.

(2) Crossing the first polar region, and bypassing next polar regions through HLL until reaching the target orbit.

⋮

(T) Crossing the first $(T-1)$ polar regions, and bypassing another polar region through HLL until reaching the target orbit.

$(T + 1)$ Crossing T polar regions and reaching the target orbit through inter-ISLs.

According to equation (3), the path length can be calculated based on the numbers of intra-ISLs and inter-ISLs. The number of inter-ISLs equals *D*. The number of intra-ISLs equals $H_C + H_L + H_T$, where H_C, H_L and H_T are defined as follow:

(1) $\boldsymbol{H_C}$: the number of intra-ISLs used to cross polar region. If t times of crossing polar region occur along the path, $H_C = t * H_{INTRA}$.

(2) $\boldsymbol{H_L}$: the number of intra-ISLs in HLL. Since intra-ISLs and inter-ISLs appear alternately and the number of appearances follow a specific pattern in HLL (see Sec. III-B), H_L can be derived if the number of inter-ISLs in HLL is known. Assume that the path has crossed polar regions for t times before arriving HLL, the orbit number when starts to follow HLL (O_L) equals $p_s + H_P * A + H_{INTER} * t * A$, where A denotes the routing direction. So the number of inter-ISLs along HLL equals $D - O_L$. H_L can be calculated based on $D - O_L$ and the appearance pattern of intra-ISLs and inter-ISLs in HLL. For example, for Starlink phase 1 shell 4 group 4 constellation, every 8 inter-ISLs corresponds to 3 intra-ISLs in HLL, so we can calculate H_L using the following formula: $1+(D-O_L)/8*3+ \mod (D-O_L, 8)/3$, where "1" represents the first intra-ISL to start following HLL.

(3) $\boldsymbol{H_T}$: the number of intra-ISLs in target orbit. Once H_C and H_L are determined, the satellite number (S_t) of the reached satellite by the packets on the target orbit (may or may not be the target satellite) can be determined as $S' - H_{INTER} * t * A + H_L * A$. H_T can be obtained by comparing S_t and S_d.

By comparing the path length of all routing candidates, the shortest path can be obtained (line 10). We use an example to illustrate how to calculate the shortest path. Assume that the source and destination satellite are $\langle 0, 0 \rangle$ and $\langle 17, 2 \rangle$, respectively. Let's firstly consider the direction with monotonically increasing orbital number, therefore, $D = 17 - 0 = 17$, $S' = (0 - 17 + 18) \mod 18 = 1$, and suppose that it takes four inter-ISLs for source satellite to reach to polar region, so $T = 1 + (17 - 4)/16 = 1$. Since $T = 1$, there are two routing options: 1) Taking 3 intra-ISLs to cross the polar region and taking inter-ISLs to reach the destination orbit, then the satellite number of the reached satellite in the destination orbit equals $(S' - 3 + 18) \mod 18 = 16$. So $H_T = 4$, and the final path contains (3+4) intra-ISLs and 17 inter-ISLs. 2) Taking HLL to reach the destination orbit. The orbit nubmer of starting following HLL is 4, therefore $H_L = 1 + (17-4)/8*3 + ((17-4) \mod 8)/3 = 5$, then the satellite number of the reached satellite in the destination orbit equals $(S' + 5) \mod 18 = 6$, and $H_T = 4$. So the final path contains $(5 + 4)$ intra-ISLs and 17 inter-ISLs. By comparing the path results of these two routing options, it is easy to find that the shortest path along the direction with monotonically increasing orbital number includes 7 intra-ISLs and 17 inter-ISLs. We can use the same method to calculate the shortest path along the direction with monotonically decreasing orbital number, and find the final path.

It is obvious that the computational complexity of the algorithm is $O(1)$. We use two parameters to represent the path: the routing direction A, and the times of crossing polar region t. These parameters are written in packets by source satellite, and the intermediate nodes along the path perform source routing following the procedure below.

2) Source Routing: If packets have arrived the target orbit, it is easy to find the routing path. Therefore, we focus on the process of how to arrive the target orbit from non-target orbits.

When a packet is received by a satellite, the path information is first extracted from the packet, i.e., the routing direction A, and the times of crossing polar region t. The satellite first checks whether the inter-ISL towards the routing direction A is on. If the inter-ISL is on, the packet is sent to the next hop through this inter-ISL. If the inter-ISL is down, it means the packet has arrived the polar edge, and it should cross the polar region or follow the HLL to bypass the polar region. In this case, the value of t is checked. If the value is greater than zero, the packet cross the polar region and update $t = t - 1$ when leaving the polar region. Otherwise the packet should follow the HLL until the target orbit is arrived.

Since source routing only needs two parameters, i.e., A and t, one byte is enough to store both parameters, one bit is used to store the A and remaining seven bits are used to store t. Considering a packet with 1500 bytes, the storage overhead introduced by source routing is negligible.

3) Handle of topological dynamics: To deal with topological dynamics, each packet will execute the Algorithm 1 to calculate the shortest path based on the current real-time topology. As evaluation result (Fig. 4) shows, the overhead introduced by Algorithm 1 is negligible, since its execution time is always kept at about 40 us, regardless of the network scale.

V. Evaluation

To evaluate the performance of SER, we compare SER with OSPF in terms of the time of executing routing algorithm and length difference. We first analyze the time required for OSPF and SER to execute routing algorithms on different network scales. The network parameters used in this experiment are shown in Table I, and the latitude threshold is set to 45° N and 45° S. To compute the execution time of OSPF, we take a "snapshot" of the network, and calculate the shortest path after the link state information has been synchronized across the network.

TABLE I: Parameters of various networks

Network Size	N	M	F	θ	H_{INTER}	H_{INTRA}
256	16	16	9	53.2	11	3
512	32	16	19	53.2	12	3
1024	64	16	39	53.2	13	3
1296	72	18	45	53.2	16	3

Fig. 4 shows the result, which demonstrates that as the network scale increases, the execution time of OSPF also increases dramatically. While SER's computational complexity is $O(1)$, its execution time is always kept at about 40 us, regardless of the network scale.

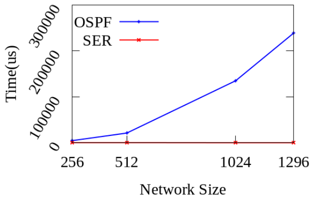

Fig. 4: The execution time of OSPF and SER with different network scale.

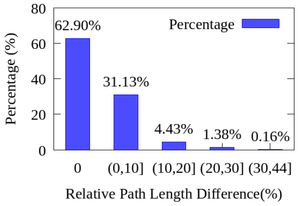

Fig. 5: The relative path length difference between SER and OSPF.

We also evaluate the length difference between the shortest paths output by OSPF and SER. We define the *Relative Path Length Difference* = (the path length of SER - the path length of OSPF)/the path length of OSPF * 100%. The experiment simulates the constellation running for one period of revolution (about 94 minutes). A snapshot of the constellation is taken every minute, SER and OSPF execute routing computing for $1296 * 1295$ source-destination pairs based on the snapshot, and the length difference rate is calculated for each source-destination pair. Fig. 5 shows the results.

The results show that 62.9% paths output by SER have the same length as that output by OSPF, 31.13% of the paths output by SER are no more than 10% longer than those output by OSPF, only 1.54% of the paths output by SER are more than 20% longer than those output by OSPF. The reason for the difference is as follows. SER assumes ploar-crossing needs three intra-ISLs. But in some cases, four intra-ISLs are required. As a result, the number of intra-ISLs is greater than the output of SER before reaching the target orbit. After reaching the target orbit, it may take one more hop through intra-ISLs in the opposite direction of the polar-crossing to reach the target satellite. In other words, if four intra-ISLs are used to cross polar region each time, two additional intra-ISLs may appear along the entire path.

VI. Conclusion

This paper presents SER, a scalable and efficient source routing mechanism for Walker Delta constellation. SER explores topology features of Walker Delta constellations, and uses only the orbit numbers and satellite numbers of source and destination to calculate the end-to-end shortest path. This routing algorithm has low computational complexity($O(1)$) and does not need to synchronize link state information across the network. SER also adopts source routing to reduce the workload of intermediate nodes along the path. Our evaluations show that SER can achieve comparable performance with that achieved by OSPF in static network settings. In future work, we will consider network failures and explore fault handling mechanism.

References

[1] Starlink. https://www.starlink.com/.

[2] OneWeb. https://oneweb.net/.

[3] Xiang Zhang et al. "ASER: Scalable Distributed Routing Protocol for LEO Satellite Networks". In: *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. 2021, pp. 65–72.

[4] Yukio Hashimoto and Behcet Sarikaya. "Design of IP-based Routing in a LEO Satellite Network". In: *Proceedings of Third International Workshop on Satellite-Based Information Services*. 1998, pp. 81–88.

[5] E. Ekici, I.F. Akyildiz, and M.D. Bender. "A distributed routing algorithm for datagram traffic in LEO satellite networks". In: *IEEE/ACM Transactions on Networking* 9.2 (2001), pp. 137–147.

[6] Thomas R. Henderson and Randy H. Katz. "On Distributed, Geographic-Based Packet Routing for LEO Satellite Networks". In: *IEEE Globecom*. 2000, pp. 1119–1123.

[7] Lixin Liu et al. "Geographic Low-Earth-Orbit Networking without QoS Bottlenecks from Infrastructure Mobility". In: *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*. 2022, pp. 1–10.

[8] Mingwei Xu et al. "Intra-domain routing protocol OSPF+ for integrated terrestrial and space networks". In: *Journal of Tsinghua University(Science and Technology)* 57.1, 12 (2017), p. 12.

[9] Zengyin Yang et al. "Analyzing and optimizing BGP stability in future space-based internet". In: *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*. 2017, pp. 1–8.

[10] Thomas Shake et al. "Contingent Routing Using Orbital Geometry in Proliferated Low-Earth-Orbit Satellite Networks". In: *MILCOM 2022 - 2022 IEEE Military Communications Conference*. 2022, pp. 404–411.

[11] DCE. https://www.nsnam.org/about/projects/direct-code-execution/.

[12] Tian Pan et al. "OPSPF: Orbit Prediction Shortest Path First Routing for Resilient LEO Satellite Networks". In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 2019, pp. 1–6.

[13] Starlink Coverage Tracker. https://starlink.sx/.

[14] Jintao Liang, Aizaz U. Chaudhry, and Halim Yanikomeroglu. "Phasing Parameter Analysis for Satellite Collision Avoidance in Starlink and Kuiper Constellations". In: *2021 IEEE 4th 5G World Forum (5GWF)*. 2021, pp. 493–498.

[15] Xiaoxin Qi, Bing Zhang, and Zhiliang Qiu. "A Distributed Survivable Routing Algorithm for Mega-Constellations With Inclined Orbits". In: *IEEE Access* 8 (2020), pp. 219199–219213.

[16] Proof of Lemma1 and Theorem1. https://github.com/qumalangma/SER/blob/main/Proof_of_Lemma1_and_Theorem1.pdf.

[17] Suzuki Ryutaro and Yasuda Yasuhiko. "Study on ISL network structure in LEO satellite communication systems". In: *Acta Astronautica* 61.7 (2007), pp. 648–658.