How to Route CUBIC and BBR Packets in Space

Shuo Huang*, Zhiyuan Wang*†‡, Wenhao Lu*, Kai Shen*, Jiayi Zhang*, Shan Zhang*†§, Hongbin Luo*†§
*School of Computer Science and Engineering, Beihang University, Beijing, China
†Zhongguancun Laboratory, Beijing, China
‡State Key Laboratory of Virtual Reality Technology and Systems, Beijing, China
§State Key Laboratory of Software Development Environment, Beijing, China

***Abstract*—Low-earth-orbit satellite constellation (e.g., Starlink) is becoming the indispensable component of future Internet. Due to the mobility nature, ground-satellite links (GSLs) and inter-satellite links (ISLs) are not as stable as the fiber links of wired Internet. Consequently, end-to-end TCP connections traversing the LEO constellation will suffer from packet loss and latency variation caused by GSL handover and ISL failure. However, typical congestion control algorithms (e.g., loss-based CUBIC and bandwidth-estimated BBR) rely on these feedbacks to detect congestion in wired Internet. This leads to performance degradation for CUBIC/BBR in LEO satellite networks. In this paper, we propose a Default-and-Backup Routing (DB-R) mechanism to enhance the performance of CUBIC/BBR in LEO satellite networks. Specifically, DB-R leverages default routing to address the packet loss caused by frequent GSL handover. Moreover, DB-R utilizes backup routing to eliminate the packet loss caused by occasional ISL failure. Overall, DB-R only incurs negligible control overhead, but can effectively improve the performance of CUBIC and BBR in LEO satellite networks. We implement DB-R in Linux kernel and evaluate its performance based on OpenSN (i.e., an open-source libarary for emulating LEO satellite networks). Results show that our DB-R mechanism can improve the average throughput of CUBIC (and BBR, respectively) by up to 37% (and 18%, respectively) compared with classic OSPF.**

***Index Terms*—LEO satellite networks, routing**

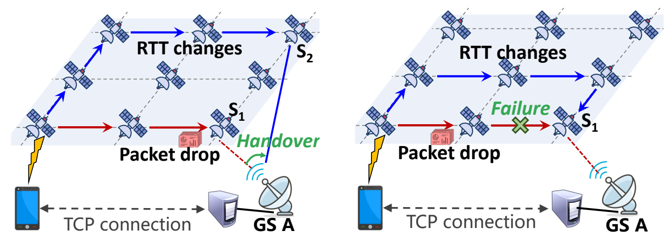
I. Introduction

A. Background and Motivation

Nowadays, low-earth-orbit (LEO) satellite constellations (e.g., Starlink [1], Iridium [2], and OneWeb [3]) have attracted increasing attentions from both industry and academia. The orbital altitude of LEO satellites is typically lower than 2000 km, leading to a smaller latency for ground-satellite links (GSLs) than geostationary-earth-orbit (GEO) satellites [4]. Recently, the technology of inter-satellite links (ISLs) is becoming mature. Many LEO constellations have been deploying ISLs [5]. The dense deployment of LEO satellites equipped with ISLs enables the LEO satellite constellation to expand the Internet services to where the terrestrial networks cannot reach [6]. Therefore, the LEO constellation is an essential part of future Internet backbone in space [7].

To unleash the potential of LEO constellations, it is necessary to enhance the existing network protocols according to the topology features of LEO constellations. Specifically, TCP/IP protocols are initially designed for terrestrial wired networks (that have relatively stable topology). Consequently, IP-based Open Shortest Path First (OSPF) protocol is used to calculate the end-to-end path, and then TCP congestion control mechanism adjusts the sending rate based on congestion signals. There have been many excellent studies on LEO satellite networking from the perspectives of intra-domain routing (e.g., host-centric [8], content-centric [9], link-centric [10]) and inter-domain routing (e.g., host-centric [11] and content-centric [12]). However, most studies above focus on the User Datagram Protocol (UDP). The studies on satellite network routing dedicated to TCP flows are still limited. The terrestrial network exhibits a stable topology, thus most TCP congestion control mechanisms presume that packet loss and latency increment can indicate network congestion. For example, CUBIC (i.e., the default in Linux protocol stack) detects network congestion according to packet loss. BBR proposed by Google in 2016 estimates the bandwidth of bottleneck links based on ACKs to update sending rate. Due to the mobility nature of LEO satellites, however, the GSLs and ISLs in LEO satellite networks are not as stable as the wired fiber links in terrestrial networks [13]–[15].

- *GSL Handover:* An LEO satellite can only provide service to the ground station (GS) within its coverage. Due to the small orbital period, GSLs will frequently handover.
- *ISL Failure:* Each satellite maintains four laser or microwave ISLs, i.e., two inter-orbit ISLs and two intra-orbit ISLs. When the relative velocity of adjacent satellites is high, the corresponding ISLs are not stable due to the antenna limitations on tracking and aiming.



(a) GSL handover (b) ISL failure

Fig. 1: Illustration of topology dynamics in LEO constellations

The aforementioned topology dynamics may lead to packet loss or RTT change. As shown in Fig. 1(a), when GS A handovers to satellite S_2 from satellite S_1, some packets may

This work was supported by National Natural Science Foundation of China under Grants 62202021, 62225201, and 62271019, in part by the National Key Research and Development Program of China under Grant 2022YFB4501000. Zhiyuan Wang is the corresponding author.

ISBN 978-3-903176-65-2

TABLE I: Related literature

Literature	GSL Handover	ISL Failure	Performance	Experiment Platform
[16]–[18]	×	×	/	Single satellite terminal
[19]	Yes (via *link-layer* notification)	×	Large overhead	Virtual network with centralized route calculation
This Paper	Yes (via default route)	Yes (via backup route)	Lightweight	Virtual network with distributed routing software

be dropped unless the routing tables are immediately updated. Moreover, the RTT of this TCP connection will suddenly change, since the route within the constellation may be different. As shown in Fig. 1(b), when a link failure occurs to an ISL, some packets may be dropped unless the routing tables are immediately updated. Even though the routing update is in time, the RTT of this TCP connection will also suddenly change. Recall that packet loss and latency are often used in typical TCP congestion control mechanisms (e.g., CUBIC and BBR). This leads to the first key question in this paper:

Question 1. *Will the topology dynamics of LEO constellation harm the performance of CUBIC and BBR?*

It is not practically feasible to revise CUBIC or BBR individually for LEO constellations. Thus, this paper aims to address the above drawbacks from another direction. Specifically, we aim to improve the routing protocol for LEO satellite networks to address the topology dynamics that may degraded the performance of CUBIC and BBR. OSPF is the routing protocol for terrestrial IP networks. It detects and disseminates the link-state changes, but does not explicitly cope with the aforementioned GSL handover and ISL failure. This motivates us to further investigate the following key question:

Question 2. *How to improve the routing protocol to mitigate negative impact of topology dynamics on CUBIC and BBR?*

To address the key questions above, we will investigate the impact of GSL handover and ISL failure on loss-based CUBIC and bandwidth-estimated BBR. Based on the obtained insights, we will further improve the routing protocol for LEO constellations to help CUBIC/BBR adapt to space environment.

B. Main Results and Key Contributions

This paper investigates the impact of GSL handovers and ISL failures on the TCP connections across LEO satellite networks. We focus on loss-based CUBIC and bandwidth-estimated BBR. Instead of tampering with classic TCP congestion control rationale, we aim to overcome the negative impact of topology dynamics via routing mechanisms, and present the design of Default-and-Backup Routing (DB-R) mechanism. Our main results and key contributions are as follows:

- *Unveiling the Impact of Topology Dynamics:* We analyze the feasibility of the congestion signals used by CUBIC and BBR, respectively. We find that GSL handover and ISL failure will inevitably cause packet loss, since the routing update is not timely enough. CUBIC senders will be misled by this packet loss as network congestion, thus reducing the congestion window. BBR senders will also be misled and underestimate the bottleneck link bandwidth, thus reducing the pacing rate. Experiments show that GSL handover and ISL failure can reduce the average throughput of CUBIC (and BBR, respectively) up to 67% (and 41%, respectively).
- *Default and Backup Routing Mechanism for LEO Constellations:* We propose a novel DB-R routing mechanism to mitigate the impact of GSL handover and ISL failure. Specifically, DB-R mechanism leverages default routing to address the packet loss caused by GSL handover. Moreover, DB-R mechanism configures backup routing to address the packet loss caused by ISL failure. Overall, DB-R mechanism incurs little overhead, but can enhance the performance of CUBIC and BBR in LEO satellite networks without modifying the end hosts.
- *Implementation in Linux Kernel:* We implement DB-R mechanism in Linux kernel 5.19.0. Specifically, we add the default routing entry to the routing table when an upcoming GSL handover is detected. We also improve the process for calculating the shortest path in OSPF, which generates backup routing entries in the routing table. Furthermore, we develop a container-based satellite network emulation platform to evaluate the performance of DB-R mechanism.
- *Performance Evaluation:* We validate the effectiveness of DB-R mechanism based on Iridium constellation on the container-based testbed. The results show that our DB-R mechanism can effectively avoid the packet loss caused by link dynamics. Moreover, DB-R reduces the average flow completion time of CUBIC (and BBR, respectively) by up to 52% (and 42%, respectively) compared to OSPF.

The rest of this paper is as follows. Section II reviews related literature. Section III introduces preliminary results. Section IV presents the design of DB-R mechanism. Section V provides experimental results. Section VI concludes this paper.

II. Literature Review

There have been many studies on congestion control mechanisms in past decades. CUBIC is loss-based, and has been the default in Linux kernel. BBR is bandwidth-estimation-based, which has been developed by Google since 2016. Now BBR is also available in Linux kernel. However, the two typical TCP congestion control mechanisms are both developed for wired networks. Hence they take the packet loss and latency increment as the congestion signal, respectively. In satellite-terrestrial integrated network (STIN), both GSLs and ISLs exhibit dynamics, leading to additional packet loss and latency variation. To this end, the two congestion control mechanisms

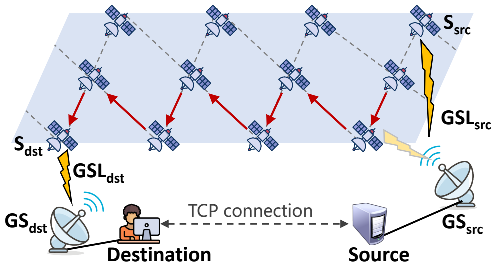


Fig. 2: TCP connection via LEO constellation

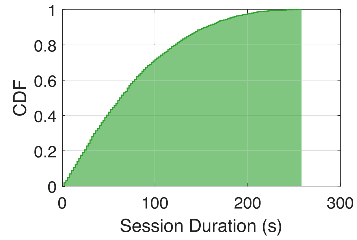


Fig. 3: Distribution of session duration without GSL handover

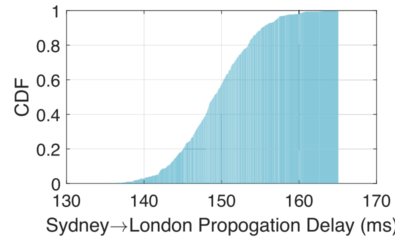


Fig. 4: Distribution of propagation delay across different ISLs

may not fit STIN. Next, we focus on the studies on congestion control in STIN. Table I summarizes the related literature.

Some studies (e.g., [16]–[18]) have been investigating the feasibility (and the improvement) of CUBIC and BBR in STIN. For example, Caini *et al.* in [16] propose TCP Hybla to address the problem of large RTT in STIN. Claypool *et al.* in [17] evaluate the performance of CUBIC and BBR based on a single satellite. Petrou *et al.* in [18] evaluate the performance of transport protocols under Sliding Windows Forward (SWF) error codes over a satellite communication link. However, the above studies focus on the time-varying propagation delay in STIN, and do not consider the impact of GSL handover. Moreover, their experiment platforms are based on real satellite terminals and a single LEO/GEO satellite. Hence they also overlook the topology dynamics caused by ISL failure and recovery.

As far as we know, Cao *et al.* in [19] first take into account the impact of GSL handover on classic congestion control mechanisms. They propose a link-layer notification scheme to address this challenge based on the predictable satellite trajectory. They also implement the proposed method in Linux kernel and evaluate the performance based on a Mininet-based virtual network environment. Despite the performance improvement, this work overlooks the following two aspects. First, the link-layer notification incurs additional overhead, and requires the terrestrial end hosts to modify the congestion control mechanisms based on the link-layer signals. Such a control overhead is significant, especially when the number of GSes is large. Second, the testbed in [19] does not support distributed routing software, but relies on centralized routing calculation as the alternative. Such a simplification hides the impact of link-state changes in terms of routing reachability.

This paper proposes Default-and-Backup Routing (DB-R) mechanism to enhance the performance of CUBIC/BBR in LEO satellite networks. Specifically, DB-R is lightweight and can address the packet loss caused by GSL handover and ISL failure effectively. We implement DB-R in Linux kernel and evaluate its performance based on OpenSN (i.e., an open-source library for emulating LEO satellite networks) [20], which runs distributed routing software FRR [21]. As far as we know, this is the first study that overcomes the impact of GSL handover and ISL failure on CUBIC and BBR.

III. Preliminary Results

A. Transmission via LEO Constellation

1) End-to-End TCP Connection: We focus on generic end-to-end TCP connections in satellite-terrestrial integrated network (STIN). As shown in Fig, 2, an end host (i.e., source) establishes the TCP connection with another end host (i.e., destination) in STIN. This TCP connection consists of the terrestrial segment and the space segment. Terrestrial segment is the path between source and GS_{src}, as well as the path between destination and GS_{dst}. Both are stable in practice. Space segment consists of two GSLs and multiple ISLs, which are intermittent and dynamic links. Therefore, the link dynamics on GSL and ISL will affect the end-to-end TCP connection. Next we introduce the details.

2) GSL Handover: Due to the mobility nature of LEO satellites, the satellite connected to a GS will change over time. Roughly speaking, the pass duration of an LEO satellite depends on the orbital altitude h and the elevation angle β. We take Starlink (i.e., $h = 550$ km, $\beta = 25°$) as the example. Fig. 3 and Fig. 4 show the empirical distributions obtained from trajectory emulations under Starlink Shell-I constellation. Fig. 3 shows the distribution of the TCP session duration without GSL handover, where the two end hosts are located in Sydney and London, respectively.

3) ISL Failure: The high-speed movement of LEO satellites renders the relative positions change constantly, which would lead to disconnection and re-connection for ISLs. ISL failures would change the end-to-end path within the constellation, thus changing the end-to-end delay. We take the route from Sydney to London as the example. Fig. 4 shows the distribution of end-to-end propagation delay across different inner-constellation paths.

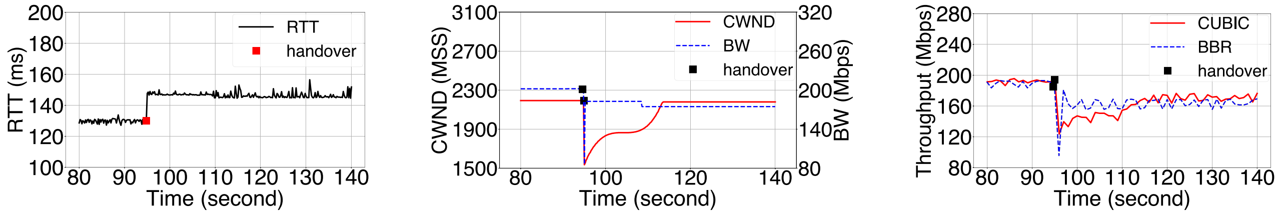
B. CUBIC in LEO Constellation

1) Rationale of CUBIC: CUBIC is a loss-based congestion control mechanism. It adjusts the sending rate according to the size of congestion window. Three duplicate ACKs trigger fast re-transmission and the congestion window reduction, i.e.,

$$\text{cwnd}_{new} = \text{cwnd}_{old} \cdot \beta, \quad (1)$$

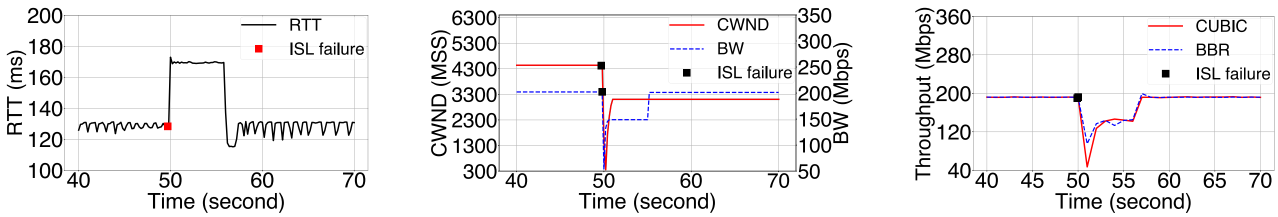
where $\beta = 0.7$ is a constant in Linux kernel. CUBIC then increases the congestion window according to the cubic function. The congestion window at time t is

$$W(t) = C \cdot (t - K)^3 + W_{\max}, \quad (2)$$



(a) RTT (b) CWND and BW (c) Throughput

Fig. 5: Impact of GSL handover



(a) RTT (b) CWND and BW (c) Throughput

Fig. 6: Impact of ISL failure

where C is a constant. Parameter t represents the time elapsed since the last packet loss. Moreover, $W_{\max}$ is the window size when last packet loss occurs. K is the time required to increase the congestion window from cwnd_{new} to $W_{\max}$.

2) Impact on CUBIC: The packet loss caused by GSL handover and ISL failure will be mistakenly detected as the congestion signal by CUBIC, leading to unnecessary congestion window reduction. Therefore, the topology dynamics will degrade the performance of CUBIC. We take the TCP connection between Shanghai and Los Angeles via Iridium constellation as the example, and illustrate the impact of GSL handover and ISL failure on CUBIC.

Long-Lived Flow: Fig. 5 evaluates the impact of GSL handover on CUBIC based on long-lived flow. In each subfigure, the horizontal axis represents the time, and the square denotes the GSL handover. Fig. 5(a) shows that RTT suddenly increases after GSL handover. GSL handover also causes packet loss. CUBIC takes packet loss as the congestion signal, thus immediately reducing the CWND as shown by the red curve in Fig. 5(b). Later on CWND gradually increases to the regular value. As shown in Fig. 5(c), the throughput decreases a lot due to GSL handover. Fig. 6 evaluates the impact of ISL failure on CUBIC based on long-lived flow, where we configure a five-second ISL failure event. Fig. 6(a) shows that RTT suddenly increases from 130 ms to 170 ms, since the ISL failure changes the end-to-end path within the constellation. ISL failure also causes packet loss. CUBIC takes packet loss as the congestion signal, thus reducing CWND as shown by the red curve in Fig. 6(b). Consequently, Fig. 6(c) also shows that the throughput decreases a lot after ISL failure.

Empirical Flows: We also evaluate the impact of GSL handover and ISL failure on CUBIC based on the empirical flow size distribution [22]. The two sub-figures in Fig. 7 plot the flow completion time and the slowdown. Note that GSL handover enlarges the FCT and slowdown up to 5x.

C. BBR in LEO Constellation

1) Ratoinale of BBR: BBR is a bandwidth-estimation-based congestion control mechanism. It adjusts the sending rate by continuously detecting the bottleneck link bandwidth. Upon receiving an ACK, BBR sender estimates the bottleneck link bandwidth according to

$$\text{BW} = \frac{\textit{Delivered} - \textit{packet.delivered}}{\textit{Now} - \textit{packet.delivered\_time}}. \tag{3}$$

The denominator represents the time elapsed since the packet was sent, and *packet.delivered_time* represents the timestamp of the latest ACK received by the sender before sending this packet. The numerator represents the amount of data successfully transmitted during this time interval. Specifically, *Delivered* represents the amount of data successfully sent up to now, and *packet.delivered* denotes the amount of data successfully sent up to the time when this packet was sent.

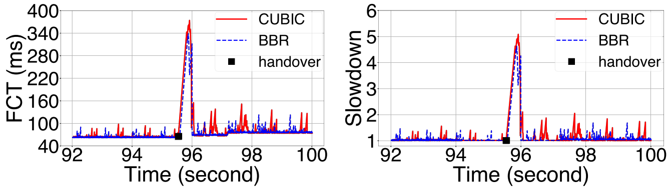
BBR takes the maximal value of BW detected within the latest ten rounds as the maximal bandwidth BW_max. BBR then calculates the pacing rate according to

$$\textit{Pacing\_rate} = \textit{BW\_max} \cdot \textit{pacing\_gain}, \tag{4}$$

where *pacing_gain* is periodically changing and takes values in $\{1.25, 0.75, 1, 1, 1, 1, 1, 1, ...\}$.

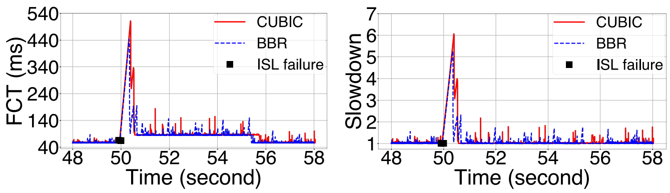
2) Impact on BBR: The packet loss caused by GSL handover and ISL failure renders BBR underestimate BW_max, which affects the sending rate of BBR sender. Similarly, we take the TCP connection between Shanghai and Los Angeles via Iridium constellation as the example, and illustrate the impact of GSL handover and ISL failure on BBR.

Long-Lived Flow: Fig. 5 evaluates the impact of GSL handover on BBR based on long-lived flow. Specifically, Fig. 5(a) shows that RTT suddenly increases when GSL handover occurs. Accordingly, BBR will underestimate the bottleneck link bandwidth. As shown by the blue curve in



(a) FCT (b) Slowdown

Fig. 7: Impact of GSL handover on FCT and slowdown



(a) FCT (b) Slowdown

Fig. 8: Impact of ISL failure on FCT and slowdown

Fig. 5(b), BW immediately drops to 80 Mbps, and then gradually increases to 180 Mbps. As shown in Fig. 5(c), the throughput also decreases due to GSL handover. Fig. 6 evaluates the impact of ISL failure on BBR based on long-lived flow, where we configure a five-second ISL failure event. As shown in Fig. 6(a), the ISL failure changes the end-to-end path, thus the RTT increases from 130 ms to 170 ms. Based on this feedback, BBR will underestimate the bottleneck link bandwidth. As shown by the blue curve in Fig. 6(b), BW immediately drops to 50 Mbps, and then gradually increases to 150 Mbps. Consequently, Fig. 6(c) shows that the throughput also decreases when ISL failure occurs.

Empirical Flows: We also investigate the impact of GSL handover and ISL failure on BBR based on empirical flow size distribution [22]. Fig. 7 indicates that GSL handover increases the FCT and slowdown up to 4.9x. Fig. 8 shows that ISL failure enlarges the FCT and slowdown up to 5x.

IV. Design of DB-R Mechanism

This section presents the design of our proposed Default-and-Backup Routing (DB-R) mechanism. Specifically, we first present the default routing mechanism for GSL handover in Section IV-A. We then present the backup routing mechanism for ISL failure in Section IV-B.

A. Default Routing for GSL Handover

1) Packet Loss Caused by GSL Handover: GSL handover usually causes packet loss in satellite networks. When a GSL is disconnected, the satellite's routing entries that correspond to the interface of the disconnected GSL will immediately disappear. This triggers the satellite to recalculate the route and then update its routing table. During this process, there are still some packets reaching this satellite along previous path, since the link-state announcement regarding the disconnected GSL also takes time. Consequently, the arriving packets will be dropped, since there is no matched routing entry. As shown

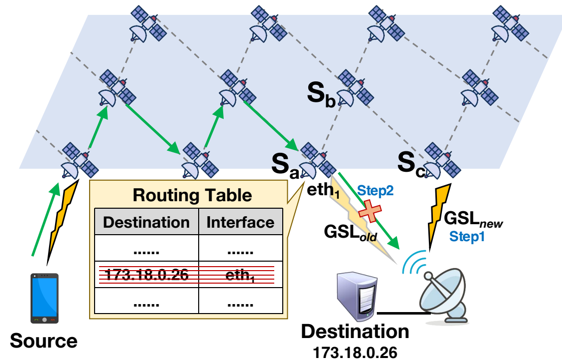


Fig. 9: Illustration of packet loss caused by GSL handover

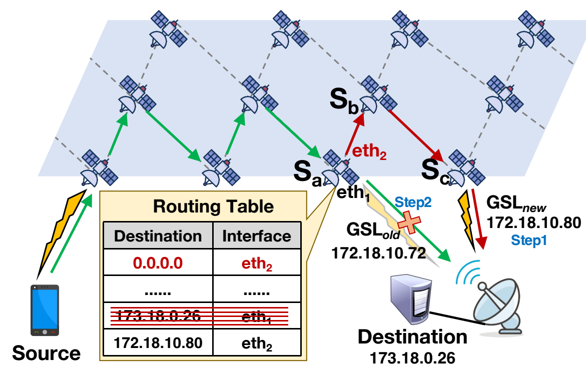


Fig. 10: Illustration of default routing for GSL handover

in Fig. 9, satellite S_a is the satellite corresponding to the disconnected GSL_{old}. The LSA generated by disconnected GSL_{old} will trigger routing update within the constellation. After routing convergence, packets sent by the source will reach the egress satellite S_c. Before that, some packets will be forwarded to satellite S_a along the previous path, i.e., the green arrows in Fig. 9.

Despite the packet loss, the time period when the routing entry is missing will not be long. According to our experiments on virtual network emulation environment, such a period lasts for about tens of milliseconds. This motivates us to eliminate the packet loss caused by GSL handover by configuring default routing at a low cost. Next let us introduce the key idea.

2) Key Idea of Default Routing for GSL Handover: By definition, default routing is a node's default choice when it is going to forward packets but finds no routing entry matched. We leverage default routing to address the temporarily missing routing entry caused by GSL handover. Specifically, when a satellite is going to disconnect the old GSL, it will add a default routing entry to its routing table. After the old GSL disconnects, this default routing entry is used to guide the packets to the new GSL. Hence we need to address the following two issues:

- When should the satellite configure default routing?
- How should the satellite configure the outgoing interface of the default routing entry?

3) Handover Detection: To achieve seamless GSL handover, the GS usually establishes a new GSL with a latter satellite before disconnecting the old GSL with the former satellite [23]. The establishment of a new GSL is actually a signal of the upcoming GSL handover. It could be viewed as the trigger event of configuring default routing for the former satellite. To proceed, we need the following preparations:

- The IP addresses of all GSLs associated to the same GS have the same IP address prefix.
- Each satellite maintains a GSL Array that records the IP addresses of all the GSLs in use for this satellite.

Given the above preparations, if a new routing entry appears in a satellite's routing table, then the satellite will check whether the destination address of the new routing entry has the same prefix as any IP address in its GSL Array.

- If yes, then this means that the satellite has detected the trigger event of configuring default routing. It would immediately add a default routing entry in its routing table (to be introduced in Section IV-A4).
- If no, then this new routing entry is not the trigger event of configuring default routing for this satellite.

As shown in Fig. 10, the IP addresses of GSLs connected with GS have the same prefix of 172.18.10.64/26 and the IP address of the GSL_{new} is 172.18.10.80. The event sequence in this scenario is as follows:

- GS establishes GSL_{new} with satellite S_c. The LSA associated with newly established GSL_{new} is disseminated from satellite S_c to other satellites.
- Satellite S_a recalculates the route upon receiving this LSA. A new routing entry with the destination address of 172.18.10.80 will be added to its routing table.
- Satellite S_a finds that the destination address of the new routing entry has the same prefix as one of IP addresses in its GSL Array. Then satellite S_a would immediately configure default routing, and disconnect GSL_{old} with GS.

4) Interface Selection: Upon detecting the trigger event of configuring default routing, the satellite needs to determine the outgoing interface of the default routing entry. Specifically, the satellite associated with the old GSL will calculate the shortest path towards the satellite associated with the new GSL. As shown in Fig. 10, upon receiving the LSA associated with GSL_{new}, satellite S_a will recalculate the shortest path towards satellite S_c. Then a new routing entry with the outgoing interface of $\mathtt{eth}_2$ will be loaded into the routing table of satellite S_a. When satellite S_a detects the newly added routing entry meets the condition specified in Section IV-A3, it would configure a default routing entry with the same outgoing interface as the new routing entry, i.e., $\mathtt{eth}_2$.

B. Backup Routing for ISL Failure/Recovery

1) Packet Loss Caused by ISL Failure: ISL failure is common in LEO satellite networks, which often results in packet loss. When an ISL becomes disconnected due to antenna misalignment, the corresponding interface of the faulty ISL become unavailable. Accordingly, the routing table entries

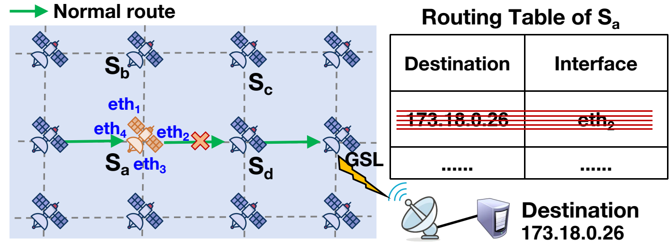


Fig. 11: Illustration of packet loss caused by ISL failure

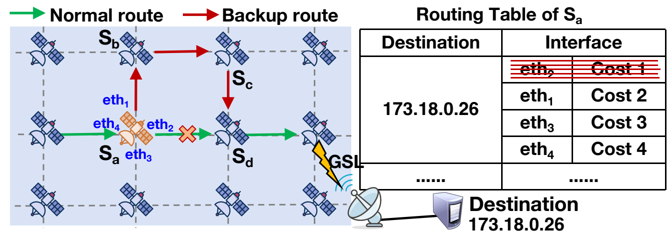


Fig. 12: Illustration of backup routing for ISL failure

involving the unavailable interface will be deleted. Then the satellite (running OSPF) will update its LSDB, recalculate the route, and disseminate the LSA for the ISL failure. Before the satellite updates its routing table, there could be some packets reaching this satellite along previous path, since LSA dissemination regarding the faulty ISL takes time. Consequently, the arriving packets will be dropped, since there is no routing entry matched. As shown in Fig. 11, satellite S_a is the satellite associated with the faulty ISL. The LSA generated by the faulty ISL will trigger routing update within the constellation. After routing convergence, packets will reach the destination along a new path. Before that, packets would still be forwarded to the satellite S_a along the previous path, i.e., the green arrows in Fig. 11.

Note that the topology of LEO constellations yields multiple equivalent path. The faulty interface caused by ISL failure is one of the possible outgoing interfaces. Hence we address the packet loss caused by ISL failure via backup routing.

2) Key Idea of Backup Routing for ISL Failure: Backup routing can significantly improve the availability of LEO satellite networks. Specifically, each satellite has four interfaces (i.e., four ISLs) that reach the destination with different costs. The interface of the smallest cost has the highest preference, and is also the main route. All the others are backup routes for this satellite. Moreover, the four entries are sorted according to their costs so that the routing entry with the minimal cost has the highest priority. This way, upon receiving a packet, the satellite could lookup the routing table to obtain the outgoing interface with the smallest cost. If some interfaces are unavailable, then the other choices are still feasible.

We illustrate the aforementioned backup routing mechanism based on Fig. 12. For satellite S_a, there are four routing entries corresponding to the destination 173.18.0.26 in its routing table. Moreover, the interface $\mathtt{eth}_2$ corresponds to the minimal cost. Hence the green arrows represent the normal path. When the ISL between satellite S_a and satellite S_d becomes faulty (i.e., the interface $\texttt{eth}_2$ of satellite S_a becomes unavailable), all the routing entries associated with the interface $\texttt{eth}_2$ immediately disappear from the routing table of satellite S_a. Before routing convergence, satellite S_a will forward the packets with destination 173.18.0.26 via the interface $\texttt{eth}_1$.

So far, we have presented how to leverage default and backup routing mechanism to address the packet loss caused by GSL handover and ISL failure. Overall, our proposed DB-R mechanism is a lightweight solution that enables CUBIC and BBR to detect the congestion more accurately in LEO satellite networks. Next we provide the experimental results in Section V.

V. Experimental Results

This section evaluates the performance of our proposed DB-R mechanism via network emulation. There have been some open-source or partially open-source LEO satellite network emulation tools, e.g., LeoEM [19], StarryNet [24], and OpenSN [20]. This paper uses OpenSN for its efficiency on constructing mega-constellations and updating concurrent network states.[1] Our experiments are based on Iridium constellation. It is a walker-star constellation that consists of 66 satellites on 6 orbital planes. The orbital altitude is 780 km. We set the bandwidth of GSL to 200 Mbps, the bandwidth of ISL to 1000 Mbps. We consider two GSes located in Shanghai and Los Angeles, respectively. Multiple TCP connections could be established via the two GSes.

In the following, we first validate the effectiveness of DB-R mechanism in Section V-A. We then evaluate the performance of DB-R in Section V-B.

A. Validation based on Long-lived Flow

We consider a single source-destination pair, and validate DB-R mechanism based on long-lived flows.

GSL Handover: Fig. 13(a) and Fig. 13(b) show the impact of GSL handover, which changes the end-to-end propagation delay. Fig. 13(a) plots the throughput achieved by CUBIC when GSL handover occurs. The blue curve corresponds to OSPF, while the red curve corresponds to DB-R mechanism. When GSL handover occurs at the 95-th second, the blue curve suddenly drops to 120 Mbps, and then gradually increases to 160 Mbps. However, the red curve directly drops to 160 Mbps due to the increment in end-to-end propagation delay. This observation indicates that DB-R mechanism enables CUBIC to adapt to GSL handover. Fig. 13(b) plots the throughput achieved by BBR when GSL handover occurs. Comparing the two curves shows that DB-R mechanism can prevent the throughput drop caused by GSL handover under BBR.

ISL Failure: Fig. 13(c) and Fig. 13(d) show the impact of ISL failure. Specifically, Fig. 13(c) plots the throughput achieved by CUBIC when ISL failure occurs, which changes the end-to-end propagation delay. In this experiment, the duration of ISL failure is 5 seconds. The blue curve corresponds to OSPF, while the red curve corresponds to DB-R mechanism. When ISL failure occurs at the 50-th second, the blue curve suddenly drops to 46 Mbps, then gradually increases to 150 Mbps. However, the red curve directly drops to 150 Mbps because of the increment in end-to-end propagation delay. This observation indicates that DB-R mechanism enables CUBIC to adapt to ISL failure. Fig. 13(d) plots the throughput achieved by BBR when ISL failure occurs. Comparing the two curves shows that DB-R mechanism can avoid the throughput drop caused by ISL failure under BBR.

[1] LeoEM [19] adopts lightweight virtualization via Mininet, thus is not capable of running distributed routing software. StarryNet [24] pays more attentions to consistency by adopting real-data-driven approach, but suffers from vertical scalability. OpenSN [20] exhibits advantages in emulation efficiency, system scalability, and function extensibility.

B. Evaluation based on Empirical Flow Size Distributions

We evaluate the performance of DB-R during GSL handover and ISL failure based on empirical flow size distribution [22]. We randomly generate flows of different sizes according to Poisson process. We focus on the average throughput, average FCT, and average slowdown during 1 second after GSL handover or ISL failure. Fig. 14 shows the results. In each sub-figure, there are four cases, i.e., CUBIC-GSL, BBR-GSL, CUBIC-ISL, and BBR-ISL. For example, the case of CUBIC-GSL represents the results running CUBIC when GSL handover occurs. Each case consists of the results achieved by classic OSPF and DB-R. Moreover, we also label the performance improvement of DB-R over the green bars.

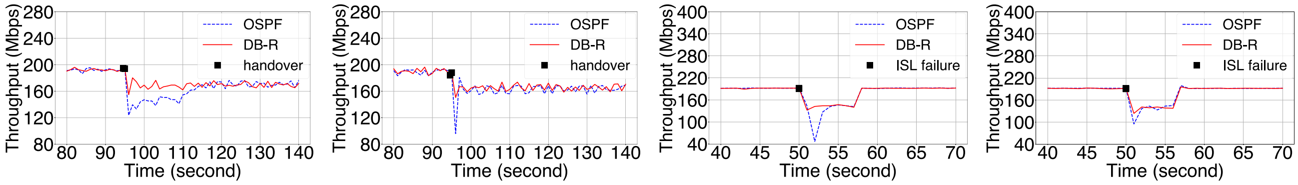
Fig. 14(a) shows the average throughput. Overall, DB-R mechanism can improve the throughput of both CUBIC and BBR when GSL handover or ISL failure occurs. Comparing the yellow bars shows that BBR outperforms CUBIC under classic OSPF. Comparing the green bars shows that CUBIC and BBR achieve almost the same performance under DB-R. This means that packet loss caused by link dynamics plays an even worse role on CUBIC than BBR. However, our proposed DB-R eliminates the impact of link dynamics, thus enabling CUBIC to perform as well as BBR in LEO satellite networks.

Fig. 14(b) plots the average FCT. Note that DB-R mechanism can reduce the FCT of both CUBIC and BBR when GSL handover or ISL failure occurs. Comparing the yellow bars shows that BBR achieves the lower average FCT than CUBIC under classic OSPF. Comparing the green bars shows that CUBIC and BBR have almost the same average FCT under DB-R. This means that link dynamics have a more significant impact on CUBIC than BBR, but DB-R can enable CUBIC to perform as well as BBR.

Fig. 14(c) shows the average slowdown. Comparing the yellow bars and the green bars shows that DB-R effectively reduces the slowdown of CUBIC and BBR by avoiding the impact of link dynamics. Hence DB-R can enhance the performance of both CUBIC and BBR in LEO satellite networks.

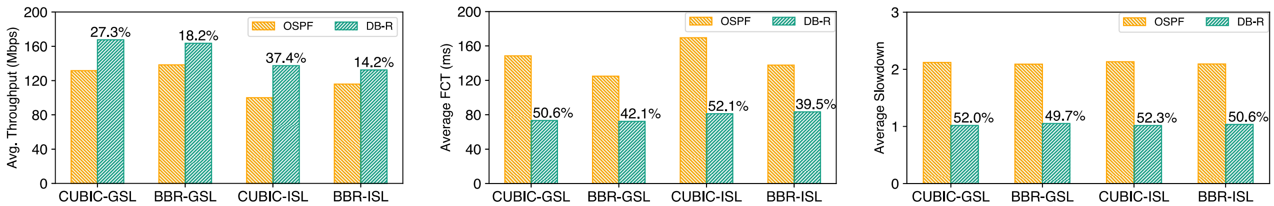
VI. Conclusion

This paper analyzes and evaluates the impact of GSL handover and ISL failure on CUBIC and BBR in LEO satellite



(a) GSL handover: CUBIC (b) GSL handover: BBR (c) ISL failure: CUBIC (d) ISL failure: BBR

Fig. 13: Validation of DB-R under long-lived flow



(a) Average Throughput (b) Average FCT (c) Average Slowdown

Fig. 14: Performance evaluation for DB-R under empirical flow size distribution

networks. Moreover, we propose DB-R mechanism to overcome the above drawbacks. Specifically, DB-R mechanism can address packet loss caused by GSL handovers by temporarily configuring default routing entry for the satellite. Moreover, DB-R mitigates packet loss due to ISL failures by generating backup routing entries. Overall, DB-R mechanism incurs little overhead, but can enhance the performance of CUBIC and BBR in LEO satellite networks without modifying the end hosts. We implement DB-R mechanism in Linux kernel 5.19.0, and validate the effectiveness of DB-R mechanism based on a container-based satellite network emulator. Experiment results show that DB-R mechanism can improve the average throughput of CUBIC (and BBR, respectively) up to 37% (and 18%, respectively) compared with classic OSPF.

References

[1] Starlink, https://www.spacex.com/webcast.

[2] IridiumNext, https://www.iridium.com/blog/iridium-next-review/.

[3] OneWeb, http://www.oneweb.world.

[4] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the internet from space with hypatia," in *ACM Internet Measurement conference*, 2020, pp. 214–229.

[5] A. U. Chaudhry and H. Yanikomeroglu, "Laser intersatellite links in a starlink constellation: A classification and analysis," *IEEE Vehicular Technology Magazine*, vol. 16, no. 2, pp. 48–56, 2021.

[6] P. Wang, J. Zhang, X. Zhang, Z. Yan, B. G. Evans, and W. Wang, "Convergence of satellite and terrestrial networks: A comprehensive survey," *IEEE Access*, vol. 8, pp. 5550–5588, 2020.

[7] G. Giuliari, T. Klenze, M. Legner, D. Basin, A. Perrig, and A. Singla, "Internet backbones in space," *ACM SIGCOMM Computer Communication Review*, vol. 50, no. 1, pp. 25–37, 2020.

[8] Q. Shan, Z. Wang, S. Zhang, Q. Meng, and H. Luo, "Routing in leo satellite networks: How many link-state updates do we need?" in *IEEE International Conference on Satellite Computing (Satellite)*, 2023.

[9] F. Yan, Z. Wang, S. Zhang, Q. Meng, and H. Luo, "Logic path identified hierarchical routing for large-scale leo satellite networks," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 4, pp. 3731–3746, 2024.

[10] H. Zhang, Z. Wang, S. Zhang, Q. Meng, and H. Luo, "Optimizing link-identified forwarding framework in LEO satellite networks," in *Proceedings of International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, 2023.

[11] E. Ekici, I. F. Akyildiz, and M. D. Bender, "Network layer integration of terrestrial and satellite ip networks over bgp-s," in *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 4, 2001, pp. 2698–2702.

[12] L. Zeng, Z. Wang, S. Zhang, Q. Meng, and H. Luo, "Adaptive interdomain content retrieval in satellite-terrestrial integrated networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2024.

[13] J. Wang, L. Li, and M. Zhou, "Topological dynamics characterization for leo satellite networks," *Computer Networks*, vol. 51, no. 1, pp. 43–53, 2007.

[14] B. Soret, I. Leyva-Mayorga, and P. Popovski, "Inter-plane satellite matching in dense leo constellations," in *IEEE Global Communications Conference (GLOBECOM)*, 2019.

[15] F. Yan, H. Luo, S. Zhang, Z. Wang, and P. Lian, "A comparative study of ip-based and icn-based link-state routing protocols in leo satellite networks," *Peer-to-Peer Networking and Applications*, vol. 16, no. 6, pp. 3032–3046, 2023.

[16] C. Caini and R. Firrincieli, "Tcp hybla: a tcp enhancement for heterogeneous networks," *International journal of satellite communications and networking*, vol. 22, no. 5, pp. 547–566, 2004.

[17] S. Claypool, J. Chung, and M. Claypool, "Measurements comparing tcp cubic and tcp bbr over a satellite network," in *IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2021.

[18] M. Petrou, D. Pradas, N. Kuhn, M. Royer, and E. Lochin, "When bbr beats fec coded cubic flows over satcom," in *Advanced Satellite Multimedia Systems Conference*, 2022.

[19] X. Cao and X. Zhang, "Satcp: Link-layer informed tcp adaptation for highly dynamic leo satellite networks," in *IEEE Conference on Computer Communications (INFOCOM)*, 2023.

[20] W. Lu, Z. Wang, S. Zhang, Q. Meng, and H. Luo, "OpenSN: An Open Source Library for Emulating LEO Satellite Networks," in *Asia-Pacific Workshop on Networking (APNet)*, 2024, pp. 149–155.

[21] FRRouting, [Online]. Available: https://frrouting.org/.

[22] P. Jurkiewicz, G. Rzym, and P. Boryło, "Flow length and size distributions in campus internet traffic," *Computer Communications*, vol. 167, pp. 15–30, 2021.

[23] B. Yang, Y. Wu, X. Chu, and G. Song, "Seamless handover in software-defined satellite networking," *IEEE Communications Letters*, vol. 20, no. 9, pp. 1768–1771, 2016.

[24] Z. Lai, H. Li, Y. Deng, Q. Wu, J. Liu, Y. Li, J. Li, L. Liu, W. Liu, and J. Wu, "StarryNet: empowering researchers to evaluate futuristic integrated space and terrestrial networks," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2023, pp. 1309–1324.