SN-FFC: Improving Survivability of LEO Satellite Network with Forward Fault Correction *

Shaoqing Wang, Youjian Zhao, Hui Xie
Department of Computer Science and Technology, Tsinghua University

ABSTRACT

Low Earth Orbit (LEO) Satellite network will suffer from heavy congestion under network faults such as node failure, which affects its survivability. Traffic engineering is the mechanism to solve such problem. In data-center wide area networks (DCWAN), Forward Fault Correction (FFC), a proactive traffic engineering system, can ensure that the network is free from congestion as long as the number of faults is up to k. Inspired by FFC, we implement SN-FFC in LEO satellite network. We show how SN-FFC can transmit data flows normally in the case of node failures. Our simulation results show that the LEO satellite network and the DCWAN have certain similarities in the fault-tolerance mechanism based on traffic engineering with FFC. Therefore, it can guide us to do more in-depth research in this area.

CCS CONCEPTS

• **Networks** → **Control path algorithms**;

KEYWORDS

LEO Satellite Network; Fault-tolerant; Traffic Engineering; Forward Fault Correction

ACM Reference Format:

Shaoqing Wang, Youjian Zhao, Hui Xie. 2018. SN-FFC: Improving Survivability of LEO Satellite Network with Forward Fault Correction . In *SIGCOMM Posters and Demos '18: ACM SIGCOMM 2018 Conference Posters and Demos, August 20–25, 2018, Budapest, Hungary*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3234200.3234252

*This work is supported by the National Science Foundation of China (No.61472214, 61472210).

SIGCOMM Posters and Demos '18, August 20–25, 2018, Budapest, Hungary

ACM ISBN 978-1-4503-5915-3/18/08. . . $15.00
https://doi.org/10.1145/3234200.3234252

1 INTRODUCTION

LEO satellite network is a way to provide broadband satellite Internet access. Many companies such as OneWeb and SpaceX's Starlink plan, plan to launch thousands of satellites to form a large satellite network, and achieve global network coverage so that more and more people can connect to the Internet.

The network organised by so many satellites will face the problem of survivability. Because the space environment is extremely complex. Moreover, it is not easy to repair the satellite fault in the space. Once the satellite node fails, the inter-satellite link will fail too. So that the entire network may suffer from heavy congestion.

In order to solve the problem of survivability, we drew on the idea of DCWAN traffic engineering. Mainly based on the following considerations: 1) the LEO satellite network and DCWAN have the same characteristics in terms of delay and bandwidth; 2) both are belong to the individual operator and can achieve autonomous control. In particular, we are inspired by the FFC[2]. FFC is a proactive method that can ensure that there is no congestion in the network, as long as the number of network faults is not more than k.

We implement SN-FFC to solve survivability problem in LEO satellite networks. We formulate SN-FFC requirement as a linear programming problem. SN-FFC can also ensure that the entire satellite network is uncongested when the number of satellite node failures does not exceed k. Unlike FFC, SN-FFC not only faces to the problems of throughput loss and computational scalability, but also the problem of real-time movement of satellite nodes in the network. We leverage the method called "snapshot" in LEO satellite networks to overcome the satellite node's movement.

We use the simulation tool to preliminarily evaluate the performance of SN-FFC. We consider two network scales: 10 satellite nodes and Iridium constellation. For different networks, two different traffic volumes are injected and the throughput loss under different number of node failures is compared.

One of our major contributions is to discover that LEO satellite network and DCWAN have similarities in the traffic engineering-based fault tolerance mechanism, so that we can benefit with latter's methods to improve the survivability of future LEO satellite networks.

2 SN-FFC OVERVIEW

Our goal is to design a fault-tolerant mechanism to ensure that under the satellite node failure or link damage, the network can still operating normally. To this end, we introduced the idea of FFC for data-center wide area network traffic engineering and design the SN-FFC, which can guarantee that the network is free from congestion when the number of network errors does not exceed k.

SN-FFC is tunnel based forwarding. There are usually multiple paths available between ingress and egress switches in the data-center wide area network. Due to the mesh topological characteristics of the LEO satellite network, there are also multiple available paths (may not equal-cost) between source and destination nodes.

SN-FFC uses the "snapshot" idea to solve the problem of network dynamics. Snapshot routing algorithm is a typical static routing algorithm in the satellite network context. It uploads the pre-calculated routing table of each snapshot to the satellite node. Then the node can update the routing table at the snapshot switching time.

SN-FFC is formulated as a LP problem. The optimization problem is trying to minimize the loss of the network throughput, which is the same objective to FFC. SN-FFC also facing the computational challenge. To deal with this challenge, we mainly refer to "bounded M-sum" problem and sorting networks leveraged by FFC.

3 EVALUATION

We use NS2 and MATLAB to implement SN-FFC controller in the LEO satellite network. The controller is deployed in the ground. It has global information of the entire satellite network so that the TE plan can be calculated. In the experiment, we aggregate the end-to-end traffic within five minutes into a flow and assume that no more than four tunnels are available per flow (according to B4[1]). We test two network scales: 10 satellite nodes and Iridium satellite constellation. The experiment does not include a fault model, but randomly selects a certain number of failed nodes.

Table 1 lists the throughput overhead of SN-FFC with different protection levels (different value of k), which is indicated by one minus throughput ratio (throughput with SN-FFC versus without). Because two or three faults are relatively large for smaller network (10 nodes),

Table 1: Throughput overhead of SN-FFC

	SN-FFC(1)	SN-FFC(2)	SN-FFC(3)
10 nodes	13%	38%	40%
Iridium	5%	12%	19%

Table 2: TE computation time with and without SN-FFC

	without SN-FFC	SN-FFC(1)	SN-FFC(2)
10 nodes	34ms	60ms	65ms
Iridium	56ms	110ms	310ms

the SN-FFC has no no significant effect in such network compared with Iridium satellite constellation.

Table 2 shows the average computation time for SN-FFC and without SN-FFC. As we can find, in larger network scale (Iridium), a higher protection level will result in more TE computation time, which is not obvious in smaller network.

Compared with [2], our simulation results show that the LEO satellite network and the DCWAN have certain similarities in the fault-tolerance mechanism based on traffic engineering with FFC. Therefore, it can guide us to do more in-depth research in this area.

4 CONCLUSION AND FUTURE WORK

Inspired by FFC, we propose SN-FFC to improve the survivability of LEO satellite networks. SN-FFC is a proactive method. Unlike the reactive method under the faults of satellite, it can provide a flexible fault tolerance mechanism to ensure that the network is free from congestion when the number of node failures does not exceed k. However, we view our design here as merely the first step in this direction, with many promising directions for future research:

Priority issues. The vision of the future satellite network is to provide high-speed broadband access, which means that future network traffic will be prioritized. This requires the design of fault tolerance mechanism considering priority factors in the LEO satellite network.

Overlay problem. In Google's B4, traffic engineering(TE) and traditional routing are in parallel, and TE has a higher priority than traditional routing. In the future, whether the LEO satellite network adopts SDN in an one step, centralized way or a similar approach to Google? This is a question worth studying.

REFERENCES

[1] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. 2013. B4: Experience with a globally-deployed software defined WAN. In *ACM SIGCOMM Computer Communication Review*, Vol. 43. ACM, 3–14.

[2] Hongqiang Harry Liu, Srikanth Kandula, Ratul Mahajan, Ming Zhang, and David Gelernter. 2014. Traffic engineering with forward fault correction. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, 527–538.