

Distributed Probabilistic Congestion Control in LEO Satellite Networks

Pranav S. Page*, Kaustubh S. Bhargao[†], Hrishikesh V. Baviskar[‡], Gaurav S. Kasbekar[§]

Department of Electrical Engineering
Indian Institute of Technology Bombay

Email : *pranavpage@ee.iitb.ac.in, [†]kbhargao@ee.iitb.ac.in, [‡]18d070048@iitb.ac.in, [§]gskasbekar@ee.iitb.ac.in

Abstract—In a dense Low Earth Orbit (LEO) satellite constellation, using a centralized algorithm for minimum-delay routing would incur significant signaling and computational overhead. In this work, we exploit the deterministic topology of the constellation to calculate the minimum-delay path between any two nodes in a satellite network. We propose a distributed probabilistic congestion control scheme to minimize end-to-end delay, which is built on top of the existing Datagram Routing Algorithm (DRA). The decision to route packets is taken based on the latest traffic information received from neighbours. We provide an analysis of the congestion caused by a simplified DRA on a uniform infinite mesh of nodes. We compare the proposed congestion control mechanism with the existing congestion control used by the DRA via simulations, and show improvements over the latter.

I. INTRODUCTION

Dense Low Earth Orbit-based communication networks [1]–[3] have become operational and are being preferred over geostationary satellites due to the lower ground-to-satellite propagation delay [4]. Challenges faced by satellite constellations differ from those faced by terrestrial networks. Due to the dynamic and fast-moving nature of the constellation relative to the ground, association, handover and pointing between satellites are non-trivial problems [5], [6]. The inter-satellite links are characterized by high propagation and transmission delays and high bit error rates [7]. The satellites themselves have limited storage and processing capabilities [8], resulting in dropped packets in congested parts of the network. A centralized algorithm for routing would need a lot of transmissions and computation in order to send optimal paths to nodes in the network. The Datagram Routing Algorithm (DRA) exploits the geometry of the network and calculates the optimum minimum-delay path between a given pair of nodes. After that, it is the job of the congestion control algorithm to pick the next hop for a packet to reach its destination with the minimum queuing and propagation delay. The problem of choosing the next hop for the packet in the presence of congestion is the problem that this work focuses on. The choice has to be made locally, and without knowledge of the congestion level of every node along the minimum delay path. A distributed congestion control algorithm that can deal with uneven node congestion levels to route packets from source to destination with low packet drops and end-to-end delay would be easy to implement on-board, and would offer better Quality of Service (QoS).

The problem of choosing an optimum schedule that minimizes the total end-to-end delay for a given set of packets has been shown to be NP-hard in [9]. Also, the problem of deciding whether a schedule exists whose completion time is bounded by some fixed given value $k \geq 0$ is proved to be NP-complete in [10]. In [11], it is shown that approximating the minimum end-to-end delay is NP-hard. Thus, heuristic-based approaches have been used to tackle this problem. Werner et al. [12] proposed a connection-oriented routing algorithm for meeting QoS requirements, which might not be able to handle link failures well. Korçak et al. [13] uses a priority metric to decide between multiple shortest paths in the network. In [14], the authors propose running a modified Dijkstra’s algorithm on

topology snapshots. Bertsekas [15] discusses an asynchronous distributed algorithm for a broad class of dynamic programming problems that is guaranteed to arrive at the optimal estimate stored in each node as the computation proceeds infinitely. Link state broadcasting is used in [16], on which the breadth first tree search is performed. Distributed geographical routing is performed in [17], and traffic is classified according to delay sensitivity. In [18], the focus is on distributed routing and it uses a restricted flooding approach to deal with link failures. The Low Complexity Probabilistic Routing algorithm in [19] describes a probabilistic routing technique for polar orbits with a simpler priority metric, which is a form of load balancing being done on the basis of congestion level information.

To deal with sudden topology changes, some form of link-state broadcast is needed. Marcano et al. [20] propose an Ad hoc On-Demand Distance Vector routing protocol, and compare its performance with a flooding protocol to study its performance in terms of packet throughput. Luo et al. [21] introduce a refined Dijkstra’s algorithm to deal with multiple shortest paths that are generated by the classic Dijkstra’s algorithm run on the network.

Liu et al. [22] propose Selective Shunt Load Balancing (SSLB), which is a distributed congestion control algorithm, using the traffic flow as an estimate of congestion in a part of the network. Dong et al. [23] introduce a load balancing algorithm based on extended link states to deal with congestion. Zhang et al. [24] propose an Area-based Satellite Routing algorithm, which is a distributed routing protocol dividing the LEO network into areas, and thus segregating the routing decisions into inter-area routing and intra-area routing. Huang et al. [25] use a pheromone based algorithm, called the Adaptive Multipath Traffic Scheduling algorithm (AMTS), which is a strategy using pheromones as indicators of network state, which is divided into node state and link state. Zhang et al. [26] propose an on-board routing algorithm based on link state information received from neighboring nodes.

Ekici et al. [27], [28] propose the DRA which routes packets between virtual nodes based on their relative location. The DRA uses a basic threshold on the outgoing buffer to determine whether the link is congested, rather than using local congestion information. We address this problem by using the packet headers as a way of conveying traffic information in the form of a single metric indicating the congestion level, and then probabilistically choosing the next hop for a packet. We perform a mathematical analysis of the congestion control algorithm in a simple network. We simulate a typical LEO satellite constellation, and compare the performance of the DRA and our own algorithm in terms of end-to-end delay. In particular, we show that for a congested portion of the network, the probabilistic congestion control algorithm offers consistently low average end-to-end delay across flows, under high loads.

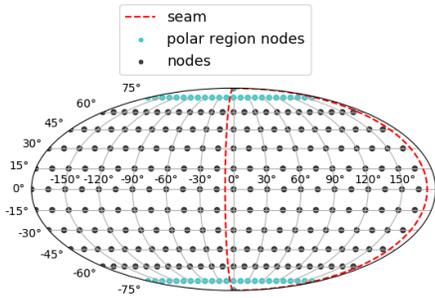


Fig. 1: Satellite constellation with $N = 12, M = 24$ and counter-rotating seam indicated

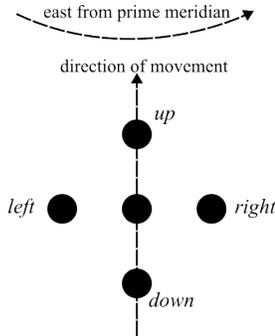


Fig. 2: Neighbors of a node with given orientation with respect to prime meridian and given direction of movement

II. SYSTEM MODEL AND PROBLEM FORMULATION

The LEO satellite constellation considered is a Walker star [29] constellation, with satellites in polar orbits. The setup and terminology used is based on the terminology used in [27]. There are N orbital planes, with M satellites per plane. Thus, the angular spacing between the planes is $360^\circ/2N$. All satellites are at a fixed altitude h from the ground, thus forming an orbital shell. We consider the network to be comprised of *virtual nodes*, as in [27], with different satellites occupying the virtual nodes at different instances of time. The virtual nodes are fixed in location with respect to the ground, and are filled up by the nearest satellite. Each virtual node location can be expressed in terms of the plane number p and the satellite number s , $0 \leq p \leq N - 1$, $0 \leq s \leq M - 1$. Each node in the network can be represented as a tuple (p, s) . This model does not deal with the mobility of satellites, and can be used to perform routing based entirely on the position of the virtual nodes, although satellites may not be at their exact locations due to orbital movement. Ekici et al. [28] provide results showing that this approximation has little effect on the end-to-end delay of packet flows, as the movement is small in magnitude. The directions of the neighbors are described in Figure 2.

Inter-plane Inter Satellite Links (ISLs) in the polar regions are considered to be shut off due to the change in the orientation of neighbors. Polar regions are defined using a latitude threshold θ_{polar} , with $\theta_{\text{polar}} = 75^\circ$ taken as the default boundary. The latitudes above θ_{polar} are considered to be in the polar regions.

Each node has four output buffers, corresponding to its four neighbors. We assume that on one link, reception and transmission can take place simultaneously. Thus, all four antennas can be transmitting and receiving simultaneously.

As seen in Figure 1, the inter-plane ISLs are shorter towards the poles and longer towards the equator. The lengths of the intra-plane ISLs are the same, as nodes are equally spaced in the orbital planes. The lengths of the inter-plane ISLs decrease with increasing latitude.

Now that the system model has been defined, the problem statement reduces to the following: Given the topology of the network of nodes and a destination node, without any global knowledge of the congestion level at each node, pick the next hop for the packet from the source node in a way that the end-to-end delay ($d_{\text{prop}} + d_{\text{queueing}}$) is minimized, where d_{prop} (respectively, d_{queueing}) is the total propagation (respectively, queueing) delay encountered by the packet while going from source to destination.

III. ALGORITHMS

The DRA consists of three phases, namely direction estimation, enhancement and congestion control. To compare the proposed congestion control algorithm with the DRA congestion control, an optimal path is obtained from the first two phases, and then the different congestion control algorithms are compared with respect to end-to-end delay. The parameters describing a path are (n_h, n_v) (number of hops in horizontal/vertical directions) and (d_h, d_v) (orientation in horizontal/vertical directions)

A. Routing

1) *DRA Direction Estimation*: In direction estimation, the path from source to destination with the minimum number of hops is chosen across paths crossing the polar region and not crossing the polar region.

2) *DRA Direction Enhancement*: This phase actually takes into account the unequal interplane ISLs in the network, and the transport of packets in the polar region. In this phase, the directions given by d_h and d_v are labeled primary or secondary, depending on the position of the packet source and destination. The primary and secondary directions are passed to the congestion control algorithm.

B. Congestion Control

1) *DRA Congestion Control*: In DRA, the decision to send packets in the primary or secondary directions is taken on the basis of the congestion level of the node's output buffers in the respective directions. If the output buffer in the primary direction has less than $N_{\text{threshold}}$ packets, then the packet is sent in the primary direction. Otherwise, if the secondary direction exists (might not exist for nodes in polar regions), and the output buffer in the secondary direction has less than $N_{\text{threshold}}$ packets, then the packet is sent in the secondary direction. If the buffers in both primary and secondary directions have buffers of size greater than $N_{\text{threshold}}$, then the packet is sent in the primary direction.

2) *New Congestion Control*: The proposed congestion control algorithm takes advantage of the dense network of nodes to transmit traffic information about a node to its neighbor. With each packet sent to its neighbor, the node adds a traffic metric to the header of the packet. Similarly, when a packet is received, the traffic metric in the header is extracted and stored in each node. Thus, each node maintains an output buffer and the latest traffic metric received for each of its neighbors. The decision to be taken involves the lengths of the output buffers as well as the traffic metric in each of the two directions.

The traffic metric to be sent has to be indicative of the congestion level of the node. The following weighted sum is used:

$$m_{node} = \frac{\sum_{i=1, i \neq j}^4 (w_{ngbr} m_{node,i} + (1 - w_{ngbr}) N_{node,i})}{3} \quad (1)$$

where m_{node} is the traffic metric to be sent from *node* to the neighbor j , $w_{ngbr} \in [0, 1]$ is the weight given to the values of the traffic metric of the neighbors of the *node*, $N_{node,i}$ is the length of the output buffer in *node* towards neighbor i , and $m_{node,i}$ is the latest traffic metric received by *node* from neighbor i .

Let *primary* and *secondary* be the primary and secondary directions given by the direction enhancement algorithm. If $N_{node,primary}$ and $N_{node,secondary}$ are the output buffer lengths in the respective directions, and $m_{node,primary}$ and $m_{node,secondary}$ are the traffic metrics received from the primary and secondary directions and stored in *node*, then the following congestion level metrics are calculated

$$c_i = w_{buffer} N_{node,i} + (1 - w_{buffer}) m_{node,i} \quad (2)$$

for $i \in \{primary, secondary\}$, $w_{buffer} \in [0, 1]$. A probabilistic distribution is chosen such that

$$P(\text{choose primary}) = f_p(c_p, c_s, p_{pref}) \\ := \frac{(c_s + 1)p_{pref}}{c_p + 1 + (c_s - c_p)p_{pref}}$$

where $c_s = c_{secondary}$, $c_p = c_{primary}$ are the congestion level metrics in the respective directions, and p_{pref} is the probability that primary is chosen when $c_s = c_p$. Similar to the DRA congestion algorithm, if $c_p < N_{threshold}$, then the packet is sent along the primary direction. If not, then the packet is sent in the primary direction with probability $P(\text{choose primary})$. The probabilistic nature of this choice ensures that a single direction is not clogged, which can happen in the DRA congestion control algorithm. Also, this ensures that the primary direction is not completely abandoned if it is clogged; the algorithm sends packets to it at a reduced rate. Preference is still given to the primary direction.

C. Mathematical Analysis

The performance of the probabilistic routing algorithm can be analyzed in a simplified network as shown in Fig. 3. We use the simplifying assumptions that the network is an infinite mesh with the property that $L_h < L_v$, and that each node is fed packets with the same rate. Each node is associated with four queues, corresponding to the four ISLs a satellite can have. For packets arriving at a node, the next hop could be in any direction. According to DRA, a primary and secondary direction is given to the congestion control algorithm, which then decides which direction to send the packet in.

As seen in the routing algorithm, if the horizontal ISLs are shorter than the vertical ISLs, even if the source-destination pair distribution is uniform, the horizontal hops can be chosen as primary more often than the vertical hops, owing to the fact that the DRA tries to have its horizontal hops in the horizontal ring where the length of the horizontal ISLs is lower than in the rings that the path might encounter in its vertical hops. We assume that p_h is the probability of choosing the primary direction to be horizontal, i.e., *right* or *left*. The choice of the secondary direction then comes down to two options,

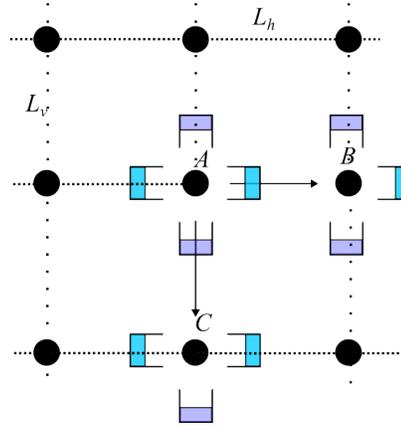


Fig. 3: Simplified mesh of nodes

orthogonal to the primary direction, as the packet does not reverse its direction. Thus, for a packet, we can assume that

$$P(\text{right primary, up secondary}) = \frac{p_h}{2} \times \frac{1}{2} \\ P(\text{up primary, right secondary}) = \frac{1 - p_h}{2} \times \frac{1}{2}$$

According to the probabilistic congestion control algorithm, node A has the latest traffic metrics from nodes B and C stored. Assuming that the traffic metric sent by *node* is purely the average of the outgoing queue lengths, i.e., $w_{neighbor} = 0$ in (1), the traffic metrics calculated by B and C to be sent to A are

$$m_{B \rightarrow A} = \frac{N_{B,right} + N_{B,up} + N_{B,down}}{3} \quad (3)$$

$$m_{C \rightarrow A} = \frac{N_{C,right} + N_{C,down} + N_{C,left}}{3} \quad (4)$$

where $N_{node,direction}$ is the instantaneous length of the queue in node *node* towards *direction*. Assuming that the mesh is infinite, and a uniform distribution of source-destination flows, the mean queue length is the same along the horizontal directions, and similarly along the vertical directions, and across nodes. Thus,

$$E(N_{B,right}) = E(N_{B,left}) = E(N_{C,right}) = N_h \quad (5)$$

$$E(N_{B,up}) = E(N_{B,down}) = N_v \quad (6)$$

Thus, the expected values of the traffic metrics are

$$E(m_{B \rightarrow A}) = m_h = \frac{N_h + 2N_v}{3}$$

$$E(m_{C \rightarrow A}) = m_v = \frac{N_v + 2N_h}{3}$$

Assuming that $w_{buffer} = 0$ in (2), the congestion levels compared are purely the traffic metrics sent by the nodes. If DRA gives *right* as the primary direction for the packet at A, and *down* as the secondary, the probability of going right is

$$P(\text{right} | \text{right primary, down secondary}) = f_p(m_h, m_v, p_{pref}). \quad (7)$$

Thus, for a packet, the probability of going right is

$$P(\text{right}) = P(\text{right primary, down secondary}) f_p(m_h, m_v, p_{pref}) \\ + P(\text{right primary, up secondary}) f_p(m_h, m_v, p_{pref}) \\ + P(\text{up primary, right secondary}) f_p(m_v, m_h, p_{pref}) \\ + P(\text{down primary, right secondary}) f_p(m_v, m_h, p_{pref})$$

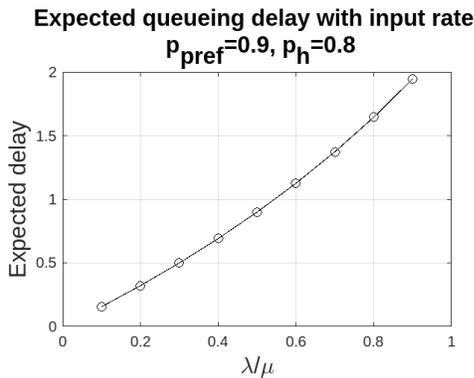


Fig. 4: Variation of normalized expected queuing delay for a path with 3 horizontal hops and 3 vertical hops with λ/μ

After simplifying, the expression reduces to

$$P(right) = \frac{p_{pref}p_h(N_v + 2N_h)}{N_h(1 + p_{pref}) + N_v(2 - p_{pref})} \frac{1}{2} + \frac{(1 - p_h)(1 - p_{pref})(N_v + 2N_h)}{N_v(1 + p_{pref}) + N_h(2 - p_{pref})} \frac{1}{2}. \quad (8)$$

Now, assuming that the input rate to node A is λ , the input rate to the queue to the right of node A is $\lambda P(right)$. Assuming that all queues are M/M/1 [30] with service rate as μ , and that all queues are stable, the mean queue length of the queue to the right of node A is

$$N_h = \frac{\rho_h}{1 - \rho_h} \quad (9)$$

where $\rho_h = \lambda P(right)/\mu$ (the number of packets in the queue follows a geometric distribution with parameter $1 - \rho$ [31]). $P(up)$ is calculated as in (8), and gives

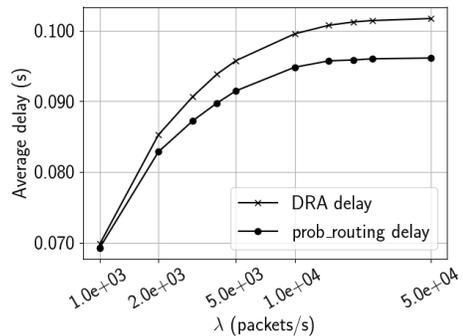
$$N_v = \frac{\rho_v}{1 - \rho_v} \quad (10)$$

where $\rho_v = \lambda P(up)/\mu$. Thus, (9) and (10) form a system with two variables, namely N_v and N_h . This system of multivariate polynomial equations was solved using MATLAB. For a path involving 3 horizontal hops and 3 vertical hops, the expected queuing delay can be expressed as $(3N_h + 3N_v)t_{tx}$, where t_{tx} is the transmission delay of a single packet, assuming that the optimal path is followed. The variation of this queuing delay (normalized with t_{tx}) with λ can be seen in Fig. 4.

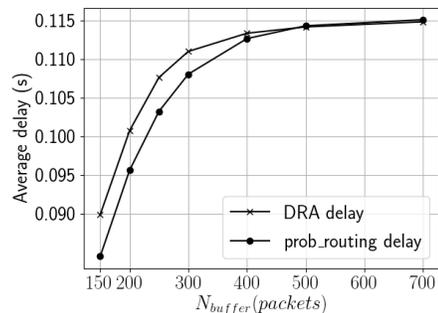
IV. SIMULATION SETUP AND RESULTS

Simulations were done in Python 3. A constellation with 12 polar orbital planes and 24 satellites per plane was used. Thus, $N = 12$ and $M = 24$ were the parameters for the constellation, with an inclination of 90° and an altitude of 600 km. A discrete event simulator was built which executed events in order of lowest time of execution. In the plot, each data point was obtained by averaging over 20 runs.

A portion of the network was simulated, bounded by the nodes (2, 3), (7, 3), (7, 9), (2, 9), consisting of 42 nodes. The simulation was fed with Poisson arrivals with rate λ_{in} in each flow. After every t_{step} seconds, n_{pairs} source-destination pairs were chosen uniformly randomly from the 42 nodes, and $n_{packets}$ packets were queued up for these pairs with exponential inter-arrival times. Then transmissions were stopped, and the network was allowed to decongest. 1 kB packets were used with a transmission rate of 25 Mbps. The buffer size was



(a) Variation with λ_{in} , for all flows



(b) Variation with N_{buffer} , for all flows

Fig. 5: Comparison of DRA and proposed congestion control algorithms

chosen to be 200, and the threshold to be 150. Parameters chosen via simulations are $p_{preference} = 0.9$, $w_{ngbr} = 0.25$, $w_{buffer} = 0.8$ to minimize average end-to-end delay.

The proposed probabilistic routing algorithm performs better than the DRA in terms of average end-to-end delay for higher input rates (λ_{in}) to the network (Fig. 5a). For $\lambda_{in} = 1.5 \times 10^4$ packets/s, the probabilistic routing algorithm gives an improvement of 5.041 ms, which is significant as the propagation delay in one hop is of the order of 6 ms. When N_{buffer} is changed, the buffer sizes in each node are increased. The ratio $N_{threshold}/N_{buffer}$ is fixed to be 0.75, to study the effect of changing N_{buffer} primarily. For large buffers, both algorithms perform similarly as no node gets congested. For lower buffer sizes, the proposed scheme performs better than the DRA in terms of end-to-end delay. Fig. 5b shows the variation of average end-to-end delay with N_{buffer} . For $N_{buffer} = 200$, the probabilistic routing algorithm gives an improvement of 5.118 ms over the DRA.

V. CONCLUSIONS AND FUTURE WORK

The proposed probabilistic congestion control algorithm offers lower average end-to-end delay in congested networks, as compared to the existing DRA congestion control algorithm. It uses the regular nature of the topology to exchange information, and effectively route packets in a probabilistic manner. For now, this scheme does not account for satellite or link failures. This could be done by flooding the network, or by incorporating a different header into the packets being sent by neighboring nodes to indicate local link failures. Loop free routing can be implemented by maintaining a partial list of the nodes visited by the packet (in (p, s) format) without significant overhead.

REFERENCES

- [1] "SpaceX non-geostationary satellite system: Attachment a technical information to supplement schedules," *US Fed. Commun. Comm., Washington, DC, USA, Rep. SAT-OA-20161115-00118*, 2016.
- [2] "Onweb non-geostationary satellite system: Attachment a—technical information to supplement schedule s," *Federal Commun. Commission, Washington, DC, USA, Tech. Rep. SAT-MOD-20180319-00022*, 2018.
- [3] S. R. Pratt, R. A. Raines, C. E. Fossa, and M. A. Temple, "An operational and performance overview of the iridium low earth orbit satellite system," *IEEE Communications Surveys*, vol. 2, no. 2, pp. 2–10, 1999.
- [4] A. Jamalipour, *Low Earth Orbital Satellites for Personal Communication Networks*. USA: Artech House, Inc., 1st ed., 1997.
- [5] F. Dosiere, T. Zein, G. Maral, and J. Boutes, "A model for the handover traffic in low earth-orbiting (leo) satellite networks for personal communications," in *Proceedings of GLOBECOM '93. IEEE Global Telecommunications Conference*, pp. 574–578 vol.1, 1993.
- [6] A. Papathanassiou and P. Mathiopoulos, "Antenna pointing algorithms for non-geostationary satellite based umts systems," in *IEEE APCCAS 2000. 2000 IEEE Asia-Pacific Conference on Circuits and Systems. Electronic Communication Systems. (Cat. No.00EX394)*, pp. 849–853, 2000.
- [7] A. H. Hashim, F. D. Mahad, S. M. Idrus, and A. S. M. Supa'at, "Modeling and performance study of inter-satellite optical wireless communication system," in *International Conference On Photonics 2010*, pp. 1–4, 2010.
- [8] M. Toyoshima, "Technological trends and future prospects of satellite communications for mega-constellations with small satellites," in *Advances in Communications Satellite Systems. Proceedings of the 37th International Communications Satellite Systems Conference (ICSSC-2019)*, pp. 1–8, 2019.
- [9] A. Clementi and M. Di Ianni, "Optimum schedule problems in store and forward networks," in *Proceedings of INFOCOM '94 Conference on Computer Communications*, pp. 1336–1343 vol.3, 1994.
- [10] D. Bovet and P. Crescenzi, "Minimum-delay schedules in layered networks," in *Acta Informatica*, vol. 28, pp. 453–461, 1991.
- [11] A. Clementi and M. Di Ianni, "On the hardness of approximating optimum schedule problems in store and forward networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 272–280, 1996.
- [12] M. Werner, C. Delucchi, H.-J. Vogel, G. Maral, and J.-J. De Ridder, "Atm-based routing in leo/meo satellite networks with intersatellite links," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 1, pp. 69–82, 1997.
- [13] O. Korcak and F. Alagoz, "Analysis of priority-based adaptive routing in satellite networks," in *2005 2nd International Symposium on Wireless Communication Systems*, pp. 629–633, 2005.
- [14] Y. Liu and L. Zhu, "A suboptimal routing algorithm for massive leo satellite networks," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–5, 2018.
- [15] D. Bertsekas, "Distributed dynamic programming," *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 610–616, 1982.
- [16] C. Liu and Y. Liu, "A real-time distributed algorithm for satellite constellation routing," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pp. 745–749, 2018.
- [17] S. Dai, L. Rui, S. Chen, and X. Qiu, "A distributed congestion control routing protocol based on traffic classification in leo satellite networks," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 523–529, 2021.
- [18] X. Qi, B. Zhang, and Z. Qiu, "A distributed survivable routing algorithm for mega-constellations with inclined orbits," *IEEE Access*, vol. 8, pp. 219199–219213, 2020.
- [19] X. Liu, Z. Jiang, C. Liu, S. He, C. Li, Y. Yang, and A. Men, "A low-complexity probabilistic routing algorithm for polar orbits satellite constellation networks," in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1–5, 2015.
- [20] N. J. H. Marciano, J. G. F. Nørby, and R. H. Jacobsen, "On ad hoc on-demand distance vector routing in low earth orbit nanosatellite constellations," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6, 2020.
- [21] Z. Luo, T. Pan, E. Song, H. Wang, W. Xue, T. Huang, and Y. Liu, "A refined dijkstra's algorithm with stable route generation for topology-varying satellite networks," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pp. 1146–1147, 2021.
- [22] J. Liu, R. Luo, T. Huang, and C. Meng, "A load balancing routing strategy for leo satellite network," *IEEE Access*, vol. 8, pp. 155136–155144, 2020.
- [23] C. Dong, X. Xu, A. Liu, and X. Liang, "Load balancing routing algorithm based on extended link states in leo constellation network," *China Communications*, vol. 19, no. 2, pp. 247–260, 2022.
- [24] X. Zhang, Y. Yang, M. Xu, and J. Luo, "Aser: Scalable distributed routing protocol for leo satellite networks," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, pp. 65–72, 2021.
- [25] Y. Huang, W. Cao, X. Liu, X. Jiang, J. Yang, and F. Yang, "An adaptive multipath routing for leo satellite network," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 4, pp. 1536–1541, 2021.
- [26] L. Zhang, F. Yan, Y. Zhang, T. Wu, Y. Zhu, W. Xia, and L. Shen, "A routing algorithm based on link state information for leo satellite networks," in *2020 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2020.
- [27] E. Ekici, I. Akyildiz, and M. Bender, "Datagram routing algorithm for leo satellite networks," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, pp. 500–508 vol.2, 2000.
- [28] E. Ekici, I. Akyildiz, and M. Bender, "A distributed routing algorithm for datagram traffic in leo satellite networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 137–147, 2001.
- [29] J. G. Walker, "Satellite Constellations," *Journal of the British Interplanetary Society*, vol. 37, p. 559, Dec. 1984.
- [30] R. Wolff, "Stochastic modeling and the theory of queues," 1989.
- [31] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, pp. 236–238. Cambridge University Press, 2013.