Distance-Based Back-Pressure Routing for Load-Balancing LEO Satellite Networks

Xia Deng, *Member, IEEE*, Le Chang, *Member, IEEE*, Shouyuan Zeng, Lin Cai, *Fellow, IEEE*, and Jianping Pan, *Senior Member, IEEE*

***Abstract*—Featuring wide coverage and high data rate, LEO satellite networks will be an important supplement to the traditional terrestrial networks, enabling the space-air-ground integrated network service. However, effective load balancing routing strategies for LEO satellite networks need to be designed, due to the bursty characteristic of the network traffic and imbalanced regional communication load. To achieve that, we propose a Distance-based Back-Pressure Routing (DBPR) strategy for LEO satellite networks. DBPR calculates the link weights based on a novel distance-based metric, which can select uncongested short-distance paths to the destinations and distribute network traffic dynamically with low delay. To control the number of forwardings in the network, we restrict the transmission range to a rectangle region between each source-destination pair. We design DBPR in the distributed fashion without collecting the global network load information, which is suitable for LEO satellite networks with limited resources, long propagation delay, dynamic topology, etc. We analyze the network stability and prove the throughput optimality of DBPR. Simulation results demonstrate that DBPR can achieve higher throughput and lower delay, compared with the state-of-the-art strategies, especially in the environments with limited cache resource.**

***Index Terms*—Back-pressure, LEO, load balancing, routing, satellite networks.**

Manuscript received 19 August 2021; revised 18 April 2022; accepted 3 September 2022. Date of publication 14 September 2022; date of current version 16 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 61702127 and 62273107, in part by the Science and Technology Program of Guangzhou under Grant 201804010461, in part by China Scholarship Council under Grants 201908440064 and 201908440085, in part by the Hundred Young Talents Plan Project of Guangdong University of Technology under Grant 220413618, in part by the Natural Sciences and Engineering Research Council of Canada, the Canada Foundation for Innovation, and in part by British Columbia Knowledge Development Fund. The review of this article was coordinated by Prof. Igor Bisio. *(Corresponding author: Le Chang.)*

Xia Deng is with the School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, Guangdong 510006, China, and also with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W3P6, Canada (e-mail: gzhu_dx@gzhu.edu.cn).

Le Chang is with the School of Automation, Guangdong University of Technology, Guangzhou, Guangdong 510006, China, and also with the Department of Computer Science, University of Victoria, Victoria, BC V8W3P6, Canada (e-mail: lechang@gdut.edu.cn).

Shouyuan Zeng is with the School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, Guangdong 510006, China (e-mail: 2111906026@e.gzhu.edu.cn).

Lin Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W3P6, Canada (e-mail: cai@ece.uvic.ca).

Jianping Pan is with the Department of Computer Science, University of Victoria, Victoria, BC V8W3P6, Canada (e-mail: pan@uvic.ca).

Digital Object Identifier 10.1109/TVT.2022.3206616

I. Introduction

With the rapid development of space technology and the increasing demands of mobile users, satellite networks have become an essential component of global mobile communication systems. Because of their global coverage and high transmission rate, satellite networks can not only provide mobile communication services between remote regions where terrestrial networks cannot reach, but also provide more reliable and faster communication services than existing wireless network technologies [1], [2], [3], [4]. Low earth orbit (LEO) satellite networks are deployed at altitudes between 160 to 2,000 KM from the earth. Comparing with other satellite networks such as geostationary orbit (GEO) and medium earth orbit (MEO) satellite networks, LEO satellite networks can achieve lower delay and higher data transmission capacity with lower power consumption, which has attracted great attention [5], [6]. For example, the Iridium system consists of 66 LEO satellites at the altitude of 780 KM, which can provide earth coverage for data services [7]. Globalstar cellular system deploys 48 satellites at the altitude of 1,400 KM to support cellular telephone systems, providing low-cost and full-coverage service [4]. In 2015, the SpaceX satellite network development plan was announced. The project aims to deploy around 12,000 small satellites to create an LEO satellite network, to work in combination with terrestrial networks in the next 20 years [8], [9].

According to the SpaceX plan, the LEO satellite network can connect to almost anywhere globally [10], [11]. Meanwhile, the LEO satellite network will further integrate with terrestrial network systems, which can support mobile broadband and machine-type communication scenarios [12], [13], [14]. Fig. 1 shows the role of LEO satellite networks in the space-air-ground integrated network, which extends the Internet coverage to sea, air and other remote areas that cannot be accessed by terrestrial wireless networks. LEO satellites also provide the service that people can transmit data from urban areas to airplanes, cruise liners and other vehicles in remote areas. Considering agriculture planting and energy mining, sensors at farms and remote mines can also be covered by satellite networks. Under the circumstances of natural disasters, such as earthquakes or floods, where the infrastructure network is damaged, satellite networks can provide the backup network service. Furthermore, satellite networks can not only develop the high-speed Internet access for rural and remote regions, but also provide competitive prices for urban areas and high quality of services for specific users [15],

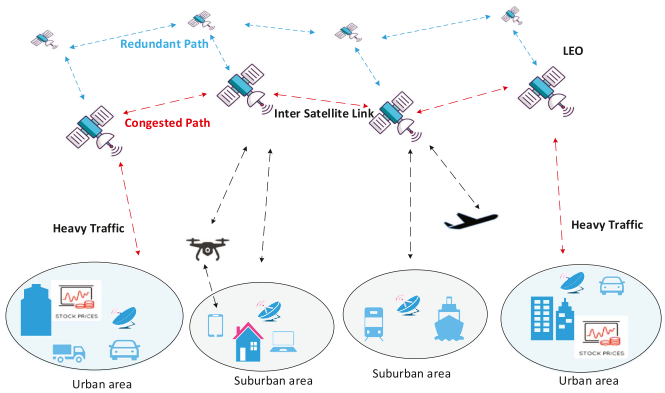


Fig. 1. The Load Balancing Problem of LEO Satellite Networks in the Space-air-ground Integrated Network.

[16]. Satellite networks can even achieve lower latency than existing optical fiber services for long-distance traffic, as the speed of light in space between satellites is 47% higher than that in optical fiber [17].

Due to the bursty characteristic of Internet traffic, the load imbalance needs to be carefully addressed in satellite networks [18], [19], [20], [21], [22], [23]. In Fig. 1, we can see that the traffic between the two urban areas is heavy and the shortest path between them maybe congested during peak hours, indicated by the red dashed arrows. However, in LEO satellite networks, there is good news. The inherent large constellations and the mesh network topology provide many candidate paths nearby, indicated by the blue dashed arrows in the figure. LEO satellite networks can offer high-speed services and low delay if the network load can be well managed and distributed on these paths. Several load-balancing routing strategies have been proposed for LEO satellite networks [24], [25], [26], [27], however, most of the above works mainly focus on designing centralized routing schemes for LEO satellite networks. The resource imbalance problem under heavy loads remains an open problem and tough task.

Back-pressure-based routing has been widely adopted to solve the load balancing problem in multi-hop networks, e.g., sensor networks and MANETs [28], [29], [30], [31], [32]. Concerning satellite networks, the links between satellites, i.e., Inter-satellite links (ISLs), also follow a multi-hop transmission manner. Therefore, the back-pressure routing strategy can potentially address the traffic dynamics and imbalance in LEO satellite networks. However, such method cannot be used in LEO networks without any change. Traditional back-pressure approaches usually explore a huge number of routing paths, and thus will suffer from large end-to-end delay and overhead under the LEO topology, which is a Manhattan network [33] with a huge number of inter-connected links and enormous available paths between sources and destinations. Some modified back-pressure routing strategies focus on reducing the delay in traditional networks [28], [29], [31]. In the shortest-path-aided back-pressure algorithm [31], packets can only be transmitted with a small hop-constraint, but the algorithm is a centralized strategy and increases the number of queues at each node, which limits its scalability. On the other hand, utilizing the sojourn time of messages may reduce the transmission delay of the messages that have been transmitted for a long time in the system [29]. However, the sojourn time cannot predict the future delay to the destinations. To summarize, these previous works can not adapt to the LEO networks, which have a Manhattan network topology and large distance between nodes.

In this paper, we further study the back-pressure routing strategy for LEO networks, with a focus on a critical metric, i.e., the distance between satellites. Different from traditional networks, such distance between nodes is very large and has a great influence on the network propagation delay in LEO networks. When determining the next hop, such distance should be considered in addition to the back-pressure strategies. We propose a distributed back-pressure routing and queue scheduling strategy to achieve adaptive resource allocation and load-aware routing, through combining the back-pressure strategy and the distance between satellites. We also restrict the transmission range for each flow, as to reduce the end-to-end delay and overhead. Such a combined approach can achieve high throughput and low delay without incurring much transmission cost. We highlight our main contributions as follows.

1) Considering the heavy load situation, we propose a novel delay metric called *Destination Distance Delay* (DDD) for LEO satellite networks, i.e., the estimated propagation delay based on the Euclidean distance from an intermediate satellite to the destination satellite. We design a Distance-based Back-Pressure Routing (DBPR) scheme where the routing metric is based on the difference of DDD in back-pressure routing, which jointly considers the queue backlog and propagation delay.
2) We analyze the network stability of the DBPR scheme using the distance delay-based metric, DDD, and prove its throughput optimality in LEO satellite networks. DBPR can achieve throughput optimality without the cost of large delay.
3) We design a distributed approach of DBPR for LEO satellite networks. Specifically, we define a constrained rectangle transmission region between the source and destination of a packet for LEO satellite networks, which limits the data transmission range and reduces transmission cost effectively.
4) We design and conduct NS2-based simulations to evaluate the performance of our proposed DBPR protocol for LEO satellite networks. Simulation results show that DBPR outperforms other related algorithms in terms of data delivery ratio and throughput, with small cost and delay.

The remainder of this paper is organized as follows. Section II introduces the related works of load-balancing routing in LEO satellite networks and back-pressure routing. Section III describes the topology model of the LEO satellite network and the preliminaries of back-pressure routing. We present our distance-based back-pressure DBPR routing scheme in Section IV. The network stability and throughput optimality of DBPR routing are analyzed in Section V. Performance evaluations are presented in Section VI, followed by concluding remarks and the discussion of further research directions in Section VII.

II. Related Works

Due to the bursty characteristic of the network traffic, the traffic distribution in LEO satellite networks is unbalanced. The load-imbalance problem is important for LEO satellite networks and has attracted great attention recently. A state-aware and load-balancing routing algorithm (SALB) was proposed in [25]. SALB estimated the link state and set the weights of the queuing delay dynamically. It considered various conditions including failure and recovery state of links and nodes, as well as the load change. An efficient shortest path tree was used to reduce the routing overhead and improve the network throughput. A hybrid global-local (HGL) load balancing routing scheme was designed in [26], where the inter-satellite traffic demand is decomposed into the predictable long-range baseline and the unpredictable short-range baseline. Network traffic can be allocated optimally, and thus the congestion can be eliminated. A minimum flow maximum residual (MFMR) routing scheme was proposed in [34]. MFMR distributed the traffic load over shortest path alternatives and kept maximum residual link capacity in satellite networks. The routing set concept and the minimization of maximum flow scheme were designed to balance the network flow. Liu et al. proposed a low-complexity routing algorithm (LCRA) [27]. LCRA adopted a distributed computation with congestion information of neighbors to select less-congested next hop to forward data. LCRA also waited for a period to forward packet when there was only one minimum next hop path to the destination. LCRA utilized traffic congestion to select the routes which improved the data delivery ratio and delay. The NCMCR routing scheme was introduced in [24], which transmitted data along multiple link-disjoint paths dynamically using the Dijkstra algorithm. NCMCR utilized network coding and the No-Stop-Wait ACK mechanism to perform continuous data transmission. The result showed that NCMCR can achieve good throughput and low end-to-end delay. Most of the above works mainly focus on designing centralized routing schemes for LEO satellite networks. Our work will design a distributed routing scheme with low cost and good scalability to solve the load-imbalance problem for LEO satellite networks.

Back-pressure-based routing has been widely applied to multi-hop networks, such as sensor networks and DTNs, to tackle the resource allocation problem [28]. Ying et al. combined the short-path information with back pressure routing to avoid unnecessarily long routes. The proposed shortest-path-aided back-pressure (SBA) algorithm achieved low end-to-end delay [31]. However, SBA is a centralized algorithm and increases the number of queues at each node, which limits its scalability. Rai et al. proposed a loop-free back-pressure routing protocol (LFBP) that forwarded packets along directed acyclic graphs (DAGs) to avoid the looping problem [30]. LFBP utilized a joint link-reversal back-pressure routing scheme and showed good delay performance. Hai et al. proposed a sojourn-time-based back-pressure routing algorithm (STBP) [29]. STBP considered the queue length and accumulated sojourn-time of packets which can reduce the delay and achieve high throughput. An on-line hybrid queue-length-based max weight scheduling algorithm (H-QMW) and adaptive A-H-QMW algorithm were proposed in [32]. They defined the capacity region for hybrid systems and designed scheduling algorithms for the coexistence of persistent and dynamic flows, which can achieve lower start-up delay and latency. Chen et al. proposed a geographic-location-aware back-pressure algorithm for data collection in a satellite network (including GEO and LEO) [35]. The algorithm used the relative rank of the distance between the neighbors and the sink, and explored short routes from the data sources to the sink on ground.

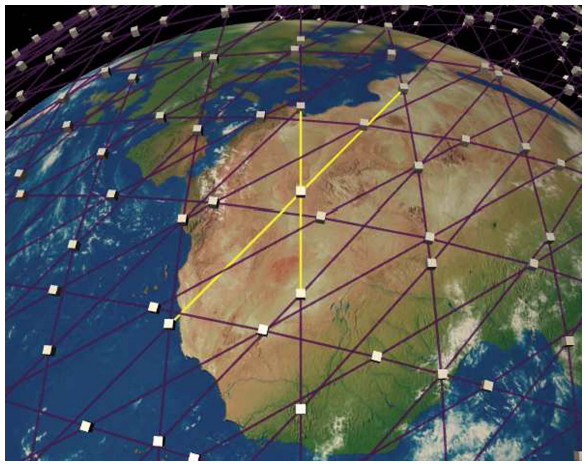


Fig. 2. Lasers Aligned for Starlink Satellites.

The above back-pressure based strategies (shortest-path, loop-free, sojourn-time, etc.) cannot apply to LEO directly. They may incur large delay and overhead due to the huge number of available paths in LEO networks. Besides, the propagation delay in LEO satellite networks is large as the distances between satellites are large. To tackle these challenges, we utilize the distance and delay in the calculation of the link weight, and restrict the transmissions within a specific area for each flow. This is different from existing works.

III. System Model and Preliminaries

A. System Model

An LEO satellite network is modeled as a graph $G = (V, E)$. V is the set of the satellites and E is the set of ISLs between satellites. There are N satellites in the network. The types of ISLs between any two neighboring satellites include intra-plane ISLs where the two satellites are in the same orbit plane and inter-plane ISLs for satellites in the neighboring orbit planes. Every node usually has two intra-plane ISLs and two inter-plane ISLs. The intra-plane ISL is used to connect neighbor nodes between its preceding and succeeding satellites in the same orbit. The inter-plane ISL is used to connect neighbor satellites in two neighbor orbits, respectively. For example, in Starlink satellite networks, the laser connections between satellites are shown in Fig. 2. The four yellow lines represent the four laser links between the satellite in the middle and its four neighbors, i.e., two intra-plane ISLs and two inter-plane ISLs.

The topology of the LEO satellite network can be viewed and modeled as a Manhattan network [33]. The 2D topology illustration is shown in Fig. 3. $S_{n,m}$ represents the m-th satellite

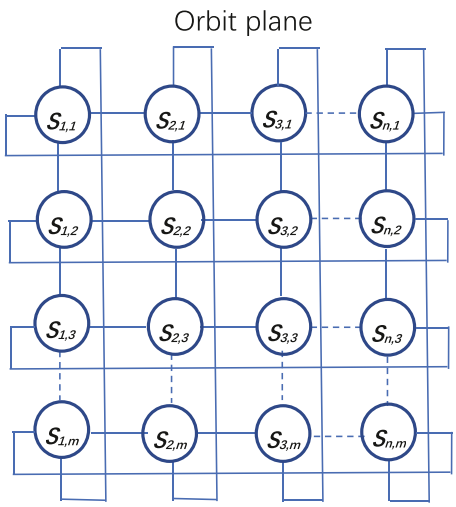


Fig. 3. Topology Model of the LEO Satellite Network.

in the n-th orbit. Every satellite has two intra-plane ISLs and two inter-plane ISLs. The length of intra-plane ISLs will not change while the length of inter-plane ISLs will. The movement of satellites is predictable and periodic. During a small time slot, we assume the topology of satellite network is fixed, which is a widely-adopted assumption for modeling satellite networks [11], [26], [36].

B. Queue-Length-Based Back-Pressure Routing

Back-pressure routing was originally proposed in [37], which is an algorithm for allocating traffic dynamically over multi-hop networks based on congestion gradients between neighboring nodes. The back-pressure routing is calculated in slotted time and the routes are determined during each time slot which can maximize the differential backlog in the network. Given a satellite network $G = (V, E)$, the link between satellite a and satellite b is denoted as link (a, b). $Q_a^c(t)$ is the queue backlog (length) of flow c in node a at time slot t. $Q_b^c(t)$ is the queue backlog of flow c at node b at time t. The differential backlog for flow c between node a and b is $Q_a^c(t) - Q_b^c(t)$. Considering there are many flows at a node, we define the weight $W_{ab}(t)$ as the largest differential backlog for all the flows on link (a, b).

$$W_{ab}(t) = \max_{c:(a,b)} [Q_a^c(t) - Q_b^c(t)]. \tag{1}$$

Then, flow c will be allocated a transmission rate over link (a, b) which is the solution to the max-weight problem as follows.

$$\text{Maximize}: \sum_{a=1}^{N} \sum_{b=1}^{N} r_{ab}(t) W_{ab}(t), \tag{2}$$

$$\text{s.t.} \quad r_{ab}(t) \in \Gamma_{s(t)}, \tag{3}$$

where $r_{ab}(t)$ is the transmission rate of link (a, b). $\Gamma_{s(t)}$ contains all possible transmission rate matrices that can be scheduled based on the network topology. According to these equations, back-pressure routing can be applied to multi-flow networks with different destinations. The routing scheme can lead to maximum network throughput and support any traffic arrival rates and channel state probabilities [38], [39].

IV. Distance-Based Back-Pressure Routing

In LEO satellite networks, the distance and propagation delay between satellites are large and we can utilize the distance between satellites to encourage packets to travel along short paths, thus improving the delay performance of LEO satellite networks. Considering the laser transmission nature in LEO satellite networks, we utilize Euclidean distance to estimate the propagation delay between satellites. In this section, we will first introduce the calculation of the Euclidean distance between satellites, and then describe our distance-based back-pressure routing algorithm.

A. Satellite Distance

Let r denote the sum of earth radius and the satellite orbit altitude. i is the inclination of the satellite orbit. Ω is the position of the ascending node, and the unit is degree. ω_1 is the angular velocity of a satellite rotating around the earth, and the unit is degree/second. γ is the initial phase angle of the satellite. ω_2 is the angular velocity of the earth rotation in degree/second. t is current time. The Euclidean distance L between any two satellites at any time t can be calculated as follows.

$$L = \sqrt{2r^2 * [1 - \cos\varphi_j \cos\varphi_k \cos(\lambda_j - \lambda_k) - \sin\varphi_j \sin\varphi_k]}, \tag{4}$$

$$\varphi = \arcsin\left(\sin i \sin\mu\right), \tag{5}$$

$$\lambda = \Omega + \arctan\left(\cos i \tan\mu\right) - \omega_2 t, \tag{6}$$

$$\mu = \omega_1 t + \gamma. \tag{7}$$

All satellites have the same orbit altitude and the same angular velocity. The length of intra-plane ISL between neighbors in the same orbit is a constant. The distance of inter-plane ISL between any two satellites in the neighboring orbit planes changes periodically. The detailed reasoning process of the formulas is described in Appendix A, and the NS-2 simulator provides a function to calculate the Euclidean distance L between any two satellites at any time t.

B. Destination-Distance-Delay-Based Back-Pressure Routing

In the traditional queue-based back pressure routing protocol, the packet backlog in a node only calculates the number of queued packets without considering the delay of a packet, which may cause the large delay problem [40], [41], [42]. The geographic distance of a packet influences the packet delay greatly, because the geographic distance between satellites is large in LEO satellite networks and it varies periodically as the satellites move. We call the satellite connecting to the destination ground station as the destination satellite. To improve the delay performance of the back-pressure routing protocol, we approximate the propagation delay of each packet in a node to the destination satellite using the Euclidean distance. Such approximated propagation delay between satellites is due to the

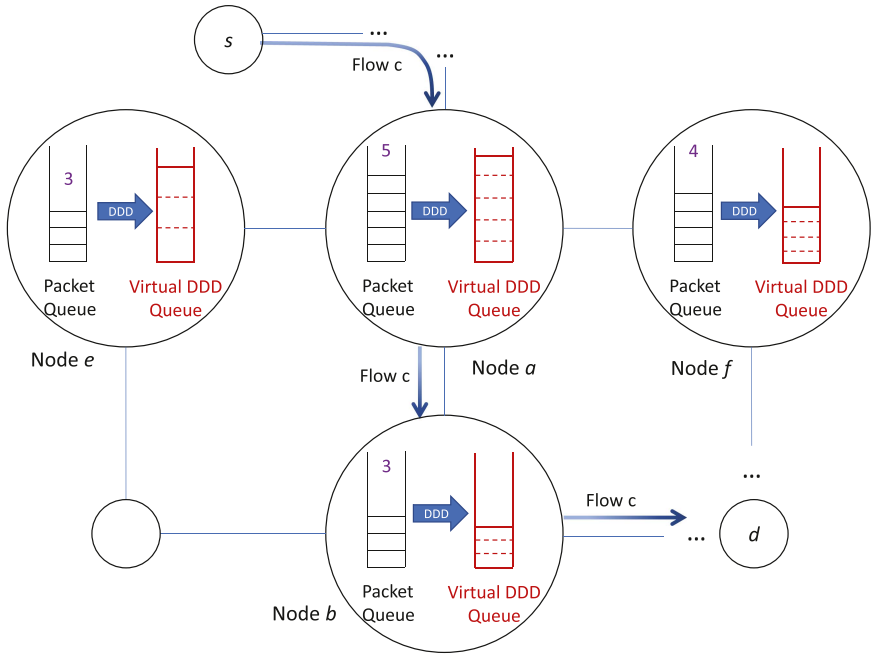


Fig. 4. DBPR Queue Management and Backlog Computation.

laser transmission nature in LEO satellite networks, referred to as *destination distance delay* (DDD) in this paper.

Let Q_a^c denote the set of packets belonging to flow c at satellite a. For each packet $p \in Q_a^c$, the Euclidean distance of the packet from satellite a to the destination satellite is denoted by $D(p)$. We use it to estimate the destination distance delay $T(p)$ of packet p by

$$T(p) = D(p)/v, \tag{8}$$

where v is the light propagation velocity in vacuum, i.e., 3×10^8 m/s, which is $1/3$ larger than that in fibers.

In LEO satellite networks, satellites move periodically and predictably. The orbit information, e.g., altitude, angular velocity, and inclination are all known, so it is easy to calculate the position of each satellite at any time. Then, according to such position information (such as longitude and latitude value) of the ground destination, the intermediate satellite can determine the corresponding destination satellite (the closest satellite to the ground destination) and calculate the distance to it in real time, i.e., $D(p)$ in Eqn. (8).

By the definition of the destination distance delay of each packet, we can obtain the cumulative destination distance delay of all the packets in node a for flow c using $\sum_{P \in Q_a^c} T(p)$. We then define the cumulative destination distance delay as a new backlog metric $\widehat{Q}_a^c(t)$ in back-pressure routing, calculated as

$$\widehat{Q}_a^c(t) = \sum_{p \in Q_a^c} T(p) = T(p) \times |Q_a^c(t)|. \tag{9}$$

$\widehat{Q}_a^c(t)$ is the destination distance delay-aware backlog of node a for flow c at time slot t. Flow c is for a specific source-destination pair. $|Q_a^c(t)|$ is the number of packets of flow c at node a at time slot t. Based on $\widehat{Q}_a^c(t)$, the differential destination distance delay-based backlog metric of link (a, b) at time slot t is defined as

$$\widehat{w}_{ab}(t) = \max_{c:(a,b)} \left[\widehat{Q}_a^c(t) - \widehat{Q}_b^c(t)\right]. \tag{10}$$

Then, flow c will be allocated a transmission rate over link (a, b) as the delay-based solution to the max-weight problem.

$$\text{Maximize}: \sum_{a=1}^{N} \sum_{b=1}^{N} r_{ab}(t) \widehat{w}_{ab}(t), \tag{11}$$

$$\text{s.t.} \quad r_{ab}(t) \in \Gamma_{s(t)}. \tag{12}$$

We name the routing scheme as Distance-based Back-Pressure Routing (DBPR) in this paper. Fig. 4 illustrates an example of the DBPR queue management and routing decision. In addition to the regular packet queue, each node maintains a virtual destination distance delay (DDD) queue to calculate the DDD-based backlog of each flow. Assume source s sends flow c to destination d. When a packet of flow c reaches node a, it will decide which node to select as the next hop as follows. The queue length of flow c at node a at time t is $Q_a^c(t) = 5$. Similarly, the queue length of flow c at node b, e and f are $Q_b^c(t) = 3$, $Q_e^c(t) = 3$ and $Q_f^c(t) = 4$, respectively. Then the differential queue backlogs of node a to b, a to e, and a to f, namely $Q_{ab}^c(t)$, $Q_{ae}^c(t)$ and $Q_{af}^c(t)$, are 2, 2 and 1, respectively. In traditional back-pressure routing algorithms, the differential queue backlogs of node a to b and a to e are the same. However, in our DBPR design, the distance to the destination will be incorporated using Eqn. (9) to make the decision. In this example, the distance from node e to the destination is greater than node b, so node b is better than node e. In addition, we assume the distances to the destination are the same for node f and b. In this case, the differential queue backlog of node a to f, $Q_{af}^c(t) = 1$,

is smaller than node a to b, $Q^c_{ab}(t) = 2$, so b is preferred. After such process, node a will select node b as the next hop.

C. Distributed Routing Algorithm

In LEO satellite networks, every satellite can connect to nearby satellites by ISLs. Due to the limited resources of satellites and the multi-hop network structure, it is suitable to design a distributed routing protocol, so we now present the details of the distributed DBPR routing strategy.

1) Queue Management: When a packet p has been transmitted and arrived at a node, the geographical Euclidean distance delay $T(p)$ between it and the destination satellite can be calculated. If it is the first packet p of flow c arriving at node n, a virtual DDD queue with $\widehat{Q}^c_n$ and $T(p)$ will be created at node n. Then $\widehat{Q}^c_n$ will be increased by $T(p)$ when packet p of flow c arrives at node n. If packet p of flow c is transmitted from node n to the next hop, $\widehat{Q}^c_n$ will be decreased by $T(p)$. The computation only involves simple incremental operations, and the complexity is $O(1)$, which is a trivial task. If the number of neighbors of a node in the network is V_n (four in this study), DBPR needs to maintain $V_n(N-1)(N-2)$ virtual DDD queues at most to record the distance-based backlog of neighbors. As there are usually a limited number of ISLs (e.g., four) for a satellite in such networks, each node needs to maintain $O(N^2)$ DDD queues, which is acceptable in terms of the buffer space and computation complexity of an individual satellite.

Concerning the communication overhead, the queue information needs to be distributed among the neighbors of a node, which can be exchanged periodically using similar approaches in other distributed networks [26], [29]. We here claim that it is possible to exchange such queue information in real time. For an LEO network consisting of 66 satellites, each satellite needs to exchange at most 65*64 $\widehat{Q}(t)$ with its neighbors, where each $\widehat{Q}(t)$ takes 4 bytes. In real applications, there is no need to exchange all the 65*64 $\widehat{Q}(t)$-s at one time. Once a new packet arrives that causes one $\widehat{Q}(t)$ to change, only the relevant $\widehat{Q}(t)$ is sent to the neighbors. Assuming each regular packet takes 512 bytes, and triggers exchanging an extra packet containing $\widehat{Q}(t)$ and such extra packet is sent to four neighbors, the overhead ratio is calculated as $4 * 4/512 = 3\%$, which is negligible in real systems for real-time $\widehat{Q}(t)$ exchange.

2) Constrained Transmission Region: Because of the grid topology of LEO satellite networks, each satellite node has four ISL links, so the number of available paths between the sources and destinations can increase at an exponential rate, depending on the number of satellites. If the intermediate satellite transmits the data through any possible links, these packets may span over a large scope of the network. Therefore, we restrict the transmissions in a rectangular area between the source and the destination to reduce the transmission redundancy, while still providing sufficient number of routing paths. As we can see from Fig. 5, for any packet from source S to destination D, the blue region is the permitted transmission region and the remaining white region is the forbidden transmission region.

In the routing process, we allocate the link with the defined largest distance delay-based weight to transfer packets within the rectangular transmission region. When a transmission is scheduled, we dequeue a packet, e.g., p, at the head of the sending buffer in node n. Then p is transmitted on link (n, b) which owns the largest $\widehat{w}^c_{nb}(t)$ among neighboring links within the permitted transmission region as

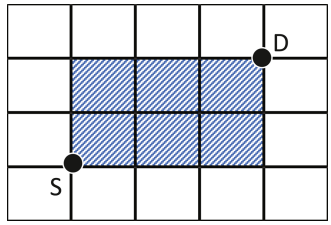


Fig. 5. The Rectangular Transmission Region between Source and Destination.

$$b = \underset{b \in N_n}{\operatorname{argmax}}\, \widehat{w}^c_{nb}(t), \tag{13}$$

$$\widehat{w}^c_{nb}(t) = \widehat{Q}^c_n(t) - \widehat{Q}^c_b(t) = \sum_{p \in Q^c_n} T(p) - \sum_{p \in Q^c_b} T(p), \tag{14}$$

where N_n are the neighbor sets of node n in the permitted transmission region. The packet of flow c will be transferred on link (n, b) at time slot t. The distributed routing algorithm of DBPR is described in detail in Alg. 1. It is worth mentioning that the algorithm is not confined to the 2-D grid topology. The differential backlog metric of links is calculated based on the destination distance delay and the number of packets of a flow, and applies to the 3-D grid topology as well, e.g., Starlink, which may further contain cross links between different shells of satellites.

V. Stability Analysis

A. Network Stability of DBPR

Assume the DBPR algorithm chooses the transmission rate $r_{ab}(t)$ and routing variable $r^c_{ab}(t)$ for link (a, b) at time slot t. The network capacity region Λ is the closure of the set of all arrival rate matrices λ^c_n for which there exists an algorithm that stabilizes the network. Stability of all queues implies that the total input rate of traffic into the network equals the total rate of data delivered to the destinations. Therefore, for any arrival rate matrix λ^c_n in the capacity region Λ, there is a stationary algorithm that chooses decision r^c_{ab} (for a long-term duration) that satisfies

$$\lambda^c_n \le \sum_{b \in N} r^c_{nb} - \sum_{a \in N} r^c_{an}. \tag{15}$$

Remark: The data rate $X^c_a(t)$ can be defined as the sum of scheduled routing variables at time slot t using

$$X^c_a(t) = \sum_{b \in N} r^c_{ab}(t). \tag{16}$$

If the routing variable $r^c_{ab}(t) = 0$, it means $r^c_{ab}(t)$ is not scheduled. Similarly, we have

$$Y^c_a(t) = \sum_{d \in N} r^c_{da}(t). \tag{17}$$

Algorithm 1: DBPR: Distance-based Back-Pressure Routing.

Enqueue a new packet:
Assume packet p was injected by flow c from source node s to destination node d; Node n receives an non-duplicate packet p
if node n is the destination of packet p **then**
Transmit p to the upper layer
else
Enqueue packet p into the buffer
Update the destination distance delay (DDD) of the packet to $T(p)$
Update the virtual DDD queue by $\widehat{Q}_n^c(t) \leftarrow \widehat{Q}_n^c(t) + T(p)$
end if
Schedule the transmission:
Dequeue the packet p at the head of the sending buffer
for all neighbors N_n of node n **do**
if N_n is in the permitted transmission region for packet p **then**
Calculate $\widehat{w}_{nb}^c(t)$ for all the permitted neighbors
end if
end for
Get the next hop b according to Eqn. (13), transmit packet p on link (n, b)
Delete packet p in the buffer
Update the virtual DDD queue by $\widehat{Q}_n^c(t) \leftarrow \widehat{Q}_n^c(t) - T(p)$

According to the stability requirement, designing a stable routing scheme is to minimize the difference between λ_n^c and $\sum_{b\in N} r_{nb}^c - \sum_{a\in N} r_{an}^c$ for an arbitrary time slot t. When the network is stable, the length of each queue should be bounded [29].

Lemma V.1: Let $\widehat{X}_n^c = \lim_{t\to\infty} \widehat{X}_n^c(t)$ and $\widehat{Y}_n^c = \lim_{t\to\infty} \widehat{Y}_n^c(t)$ be the average output and input respectively through a long-term observation. If the network is stable, the optimal transmission process satisfies

$$\lambda_n^c - \widehat{X}_n^c + \widehat{Y}_n^c \le 0. \tag{18}$$

The proof of Lemma V.1 is detailed in Appendix B.

B. Throughput Optimality of DBPR Routing

The optimal throughput scheme minimizes the change of queue backlogs from one slot to the next. We use the Lyapunov drift $\Delta(t)$ to measure the change from slot t to slot $t+1$ in DBPR routing as

$$\widehat{Q}(t) = \sum_{n\in N}\sum_{c\in N} \widehat{Q}_n^c(t), \tag{19}$$

$$\Delta(t) = \sum_{n\in N}\sum_{c\in N} E\left[\frac{1}{2}(\widehat{Q}_n^c(t+1)^2 - \widehat{Q}_n^c(t)^2)/\widehat{Q}(t)\right]. \tag{20}$$

Considering the dynamic routing process, the one-slot queue backlog satisfies

$$\widehat{Q}_n^c(t+1) \le \max\left\{\left(\widehat{Q}_n^c(t) - \widehat{X}_n^c\right), 0\right\} + \widehat{Y}_n^c + T\left(A_n^c(t)\right), \tag{21}$$

where

$$\begin{aligned}\widehat{X}_n^c(t) &= \sum_{p\in \mathbf{X}_n^c(t)} T(p) \\ &= \sum_{p\in \mathbf{X}_n^c(t)} D(p)/v,\end{aligned} \tag{22}$$

and $\mathbf{X}_n^c(t)$ are the packets from flow c transmitted along link (a, b) for all $b \in N$ at the beginning of time slot t. Similarly, $\mathbf{Y}_n^c(t)$ are the packets from flow c transmitted along link (b, a) for all $b \in N$ at the beginning of time slot t. $A_n^c(t)$ denotes the amount of the new flow c data that exogenously arrives at node n at slot t.

Note that the following inequality holds for all $q \le 0$, $a \le 0$, and $b \le 0$:

$$(\max\{q - b, 0\} + a)^2 \le q^2 + b^2 + a^2 + 2q(a - b). \tag{23}$$

By squaring $\widehat{Q}_n^c(t+1)$ and according to the above inequality, $\Delta(t)$ can be bounded as follows under any routing algorithm.

$$\begin{aligned}\Delta(t) \le B + \sum_{n\in N}\sum_{c\in N} \widehat{Q}_n^c(t) * E\left[T\left(A_n^c(t)\right)\right. \\ \left. -\widehat{X}_n^c(t) + \widehat{Y}_n^c(t)\right] / \widehat{Q}(t)],\end{aligned} \tag{24}$$

where B is a finite constant that depends on the second moments of the arrivals and the maximum possible moments of transmission rates. B and $T(A_n^c(t))$ are constants. To minimize the right-hand side of the above inequality under the routing scheme, we need to maximize

$$E\left[\sum_{n\in N}\sum_{c\in N} \widehat{Q}_n^c(t) * \left[\widehat{X}_n^c(t) - \widehat{Y}_n^c(t)\right] / \widehat{Q}(t)\right]. \tag{25}$$

By the principle of opportunistically maximizing an expectation, the above expectation is to maximize the function inside it, which is

$$\sum_{n\in N}\sum_{c\in N} \widehat{Q}_n^c(t) * [\widehat{X}_n^c(t) - \widehat{Y}_n^c(t)], \tag{26}$$

where

$$\begin{aligned}\widehat{X}_n^c(t) - \widehat{Y}_n^c(t) &= T(X_n^c(t)) - T(Y_n^c(t)) \\ &= \sum_{p\in \mathbf{X}_n^c(t)} T(p) - \sum_{p\in \mathbf{Y}_n^c(t)} T(p).\end{aligned} \tag{27}$$

We assume all the packets from the same flow arrive at a node randomly following i.i.d. at sufficiently large time t for all $n, c \in N$. The packets are scheduled by the FIFO strategy. Then we can get

$$\frac{T(X_n^c(t))}{X_n^c(t)} = \frac{T(Y_n^c(t))}{Y_n^c(t)} = \frac{T(Q_n^c(t))}{Q_n^c(t)} = \frac{\widehat{Q}_n^c(t)}{Q_n^c(t)}. \tag{28}$$

Then we maximize

$$\sum_{n\in N}\sum_{c\in N}\widehat{Q}_n^c(t)*\left[T(X_n^c(t))-T\left(Y_n^c(t)\right)\right] \tag{29}$$

$$=\sum_{n\in N}\sum_{c\in N}\widehat{Q}_n^c(t)*\left(\frac{T(Q_n^c(t))}{Q_n^c(t)}X_n^c(t)-\frac{T(Q_n^c(t))}{Q_n^c(t)}Y_n^c(t)\right) \tag{30}$$

$$=\sum_{n\in N}\sum_{c\in N}\frac{\widehat{Q}_n^c(t)}{Q_n^c(t)}*\left(X_n^c(t)T(Q_n^c(t))-Y_n^c(t)T(Q_n^c(t))\right) \tag{31}$$

$$=\sum_{n\in N}\sum_{c\in N}\frac{\widehat{Q}_n^c(t)}{Q_n^c(t)}*T(Q_n^c(t)\left[\sum_{b\in N}r_{nb}^c(t)-\sum_{a\in N}r_{an}^c(t)\right]. \tag{32}$$

Because $\frac{\widehat{Q}_n^c(t)}{Q_n^c(t)}=\frac{T(Q_n^c(t))}{Q_n^c(t)}$ is non-negative, we need to maximize

$$\sum_{n\in N}\sum_{c\in N}T(Q_n^c(t)\left[\sum_{b\in N}r_{nb}^c(t)-\sum_{a\in N}r_{an}^c(t)\right]. \tag{33}$$

By switching the sums, we obtain

$$=\sum_{a\in N}\sum_{b\in N}\sum_{c\in N}r_{ab}^c(t)\left[T(Q_a^c(t))-T(Q_b^c(t))\right] \tag{34}$$

$$=\sum_{a\in N}\sum_{b\in N}\sum_{c\in N}r_{ab}^c(t)\left[\widehat{Q}_a^c(t)-\widehat{Q}_b^c(t)\right]. \tag{35}$$

$\widehat{Q}_a^c(t)-\widehat{Q}_b^c(t)$ is the differential backlog of flow c between node a and b with the destination distance delay of the packet. The routing decision is to choose $r_{ab}^c(t)$ to maximize the above equality and minimize $\Delta(t)$, where the maximization means the routing strategy of DBPR. Then our DBPR routing scheme is an optimal throughput routing scheme by considering the distance delay between satellites.

VI. Simulation Results

A. Simulation Setup

We use the NS2 simulation platform and an Iridium-like LEO satellite network to evaluate the performance of our DBPR routing strategy. The Iridium satellite network consists of 66 satellites and each orbit has 11 satellites [7]. Each satellite has four inter-satellite ISLs, including two intra-plane ISLs and two inter-plane ISLs. Similar to the related works in LEO satellite networks [24], [27], we set homogeneous bandwidth, i.e., the up-link and down-link of satellites and all the bandwidths of ISLs between satellites are 5 Mbps. Other parameters for the Iridium satellite network are: altitude = 780 km and inclination = 86.4°. We adopt a constant traffic flow with a fixed packet size of 512 B. The simulation time is 100 seconds per run. Because the major cities on the earth and the majority of the network traffic are distributed between 60′ north latitude and 60′ south latitude. We set 8 flows (the source-and-destination pairs) distributed between 60′ north latitude and 35′ south latitude on the eastern hemisphere. The distribution of the cities and satellites involved is shown in Fig. 6, and the sources and destinations of the 8 flows are listed in Table I. A flow starts from a ground station near a city, and the ground station is connected to the LEO network through the nearest satellite. This is also the case for the destinations of flows. So the 8 flows correspond to eight pairs of source-destination ground stations (or satellites).

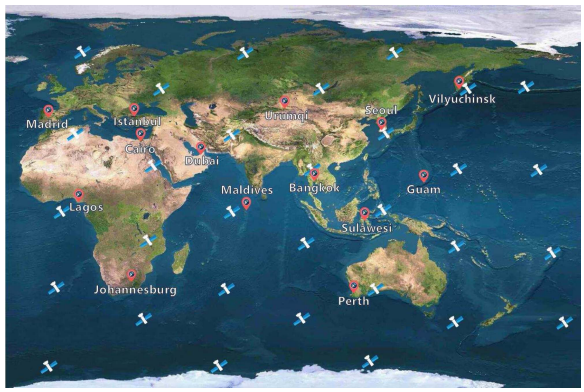


Fig. 6. Distribution of the Cities and Satellites.

TABLE I
Source and Destination of Flows

Flow number	Sources	Destinations
1	Bangkok	Madrid
2	Sulawesi	Istanbul
3	Guam	Dubai
4	Lagos	Urumqi
5	Cairo	Seoul
6	Maldives	Vilyuchinsk
7	Perth	Cairo
8	Johannesburg	Sulawesi

We implement our distributed DBPR routing algorithm in the NS2 network simulator and compare it with the SBPR protocol which utilizes the sojourn time of packets in the network instead of the queue length as the backlog in the weight calculation of the back-pressure routing protocol [29]. For a fair comparison, both DBPR and SBPR routing protocol constrain the transmissions in a rectangle region. Another algorithm is the Dijkstra-based shortest path routing protocol (SPR). Considering the overhead-balance routing protocols in the LEO satellite networks, the LCRA algorithm also obtains the routes in a distributed manner. LCRA selects the shortest paths to destinations and utilizes the congestion information of the neighboring nodes to relieve network congestion [27].

We use the following performance metrics.

1) *Data Delivery Ratio:* the ratio of the number of messages arrived at destinations to the number of messages sent by the source nodes.
2) *Average Delay:* the average time spent by all messages from the source to the destinations.
3) *Average Number of Forwardings per Packet:* the total number of forwardings over the number of messages reaching their destinations. This metric measures the average transmission cost of a delivered packet.
4) *Throughput:* the overall successful message delivery rates on the destinations.

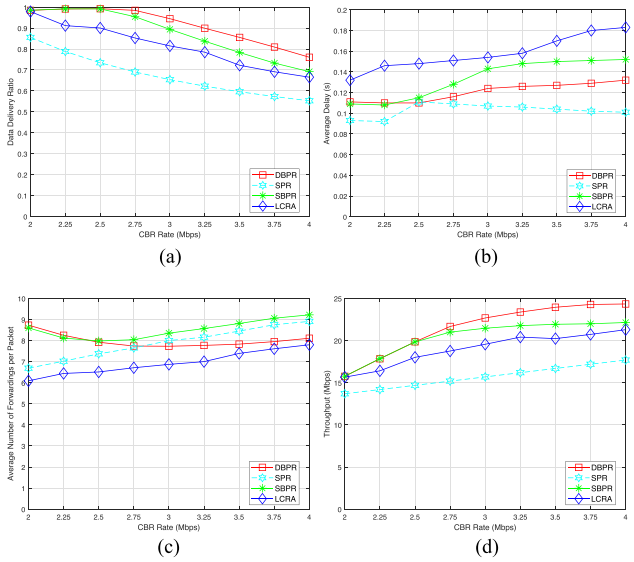


Fig. 7. The Performance with Different CBR. (a) Data Delivery Ratio. (b) Average Delay. (c) Average Number of Forwardings per Packet. (d) Throughput.

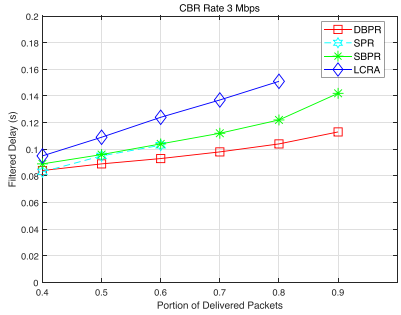


Fig. 8. The Filtered Delay with Different Portion of Delivered Packets.

B. Performance With Different Traffic Pattern

1) CBR Traffic: We vary the total CBR generation rate in the network from 2 Mbps to 4 Mbps. The buffer size of each node is 50 packets. In Fig. 7(a), when the CBR rate increases, the data delivery ratio in all protocols decreases due to the heavier transmission load. DBPR achieves the highest data delivery ratio and shows the best traffic balancing ability among these protocols because it considers both the back-pressure routing and distance delay, which can balance the network traffic without much cost on long paths or large delay. When the CBR rate is 4 Mbps, the data delivery ratio of DBPR is improved by nearly 38% comparing with SPR. The LCRA algorithm considers the neighboring node congestion information and chooses the less-congested shortest paths to destinations, which shows a higher data delivered ratio than SPR. Moreover, LCRA cannot provide adequate paths to adapt to dynamic and heavy load. It shows a lower data delivery ratio than back-pressure-based routing protocols, SBPR and DBPR, that can adapt to heavy loads dynamically. Different from SBPR which utilizes the sojourn-time of packets, DBPR adopts the approximated delay from intermediate satellites to destination satellites as the link weights, which can balance the congestion and distance flexibly using destination distance delay-based differential backlog between neighboring nodes. The data delivery ratio results show that DBPR achieves the highest data delivery ratio when the network traffic becomes heavy.

We plot the average delay performance in Fig. 7(b). The shortest path protocol SPR shows the lowest average delay. This is because SPR always selects the shortest paths and a large number of packets will be dropped when the load is heavy. Our scheme DBPR also performs well in terms of average delay. It is lower than LCRA and SBPR. It verifies that DBPR can balance the transmission delay and queuing delay efficiently, and achieve an overall low delay. Because LCRA postpones sending data when the link is congested and only selects the next hop from no more than two neighbors, it achieves the highest delay.

To compare the delay performance of strategies with different data delivery ratio further, we set the same portion of delivered packets and investigate the average delay of these selected packets. For example, if the delivery ratio of strategy A is 70% and B is 80%, we can safely claim that strategy B is better than A in terms of delivery ratio. However, it is difficult the compare the delay performance, because the numbers of delivered packets are different. To make it a fair comparison, we need to pick up 70% of all the packets from the 80% delivered packets in strategy B. We do this by selecting and calculating the average delay of the first 70% of the delivered packets in B in the ascending order of delay and refer to such delay the filtered delay. By doing so, we can compare the average delay of strategy A and strategy B under the same data delivery ratio, i.e., 70%. The CBR rate is 3 Mbps. The filtered delay of packets under given portions of delivered packets is plotted in Fig. 8. Different from Fig. 7(b) where SPR owns the lowest average delay, we observe that DBPR reaches the lowest filtered delay in Fig. 8. That is because SPR demonstrates the smallest data delivery ratio (only 0.66) and drops a large number of packets. For the same portion of arrived packets, such as 0.5 and 0.6, DBPR shows a lower delay than SPR. DBPR can even achieve a data delivery ratio higher than 0.9. Therefore, we can infer that there exists some packets delivered to the destinations with a larger delay than the shortest-path delay, but these packets delivered to the destinations help increase the data delivery ratio. From Fig. 8, we can conclude that our protocol DBPR achieves the best average delay.

The performance on the average number of forwardings per packet is shown in Fig. 7(c), which indicates the transmission cost of the compared strategies. Through restricting the permitted transmission area to a fixed range between the source and destination of all packets, DBPR shows a low average number of forwardings per packet. Although the network topology of the LEO satellite network is a Manhattan-based network that provides lots of redundant paths, we can see that restricting the transmission within a certain area enables our proposed scheme DBPR to achieve a small transmission cost, without compromising the data delivery performance. Comparing with SPR, DBPR achieves a lower cost when the CBR rate exceeds 2.75 Mbps. LCRA shows the lowest number of forwardings as it selects the next hop between no more than two shortest routes

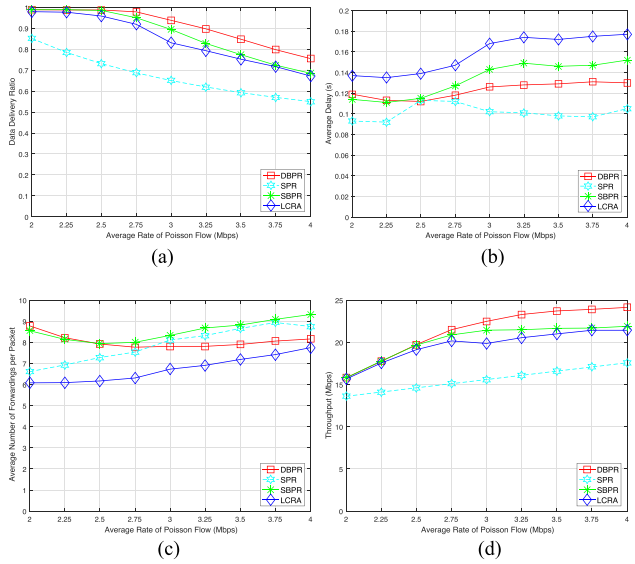


Fig. 9. The Performance under Poisson Traffic. (a) Data Delivery Ratio. (b) Average Delay. (c) Average Number of Forwardings per Packet. (d) Throughput.

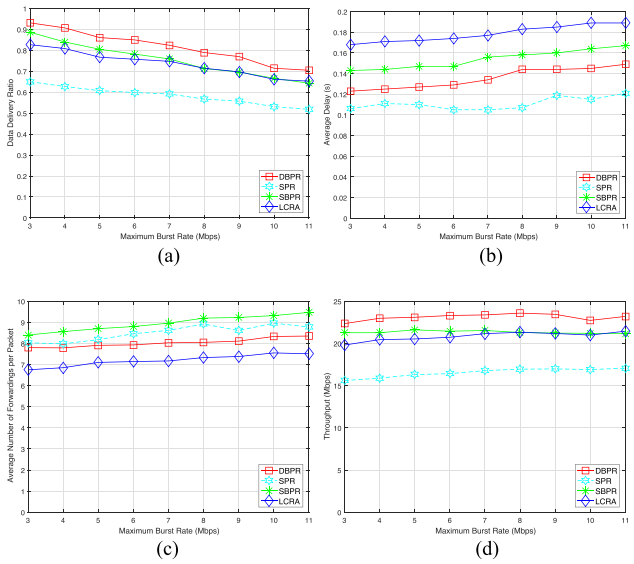


Fig. 10. The Performance under Bursty Traffic. (a) Data Delivery Ratio. (b) Average Delay. (c) Average Number of Forwardings per Packet. (d) Throughput.

and considers the congestion information of neighboring nodes. However, when the CBR rate is close to 4 Mbps, DBPR shows a similar performance with LCRA. Therefore, we can see that our scheme still owns a low transmission cost in the circumstance of heavy traffic. SBPR, which considers the sojourn time of packets and selects more redundant route paths to destinations, shows the largest transmission cost among these protocols.

In Fig. 7(d), the throughputs of all the protocols increase with more CBR flows injected into the network. We can see that DBPR can achieve the largest throughput and SPR shows the smallest. LCRA performs lower than back-pressure-based routing SBPR and DBPR as it selects the next hop between two routes which are insufficient when the traffic becomes heavy. It verifies that our proposed DBPR can always improve the network throughput.

Based on the performance in Fig. 7, we conclude that the DBPR scheme can achieve the highest data delivery ratio and throughput without introducing a large delay or transmission cost. Specifically, it demonstrates a high balancing ability when the network load is heavy.

2) Poisson Traffic: We also evaluate the performance of the strategies under Poisson traffic at different bit rates. The results are shown in Fig. 9, similar to the performance with the CBR flows in Fig. 7. DBPR achieves the best data delivery rate and throughput. It is also better than SBPR and LCRA in terms of delay, and better than SPR and SBPR in terms of transmission overhead. In summary, DBPR demonstrates a desirable comprehensive performance in terms of the four evaluating metrics.

3) Bursty Traffic: The performance under bursty traffic is shown in Fig. 10. The entire process consists of multiple data transmission periods, and each period is composed of two phases: the normal phase and the bursty phase. Each normal phase lasts 2 seconds on average with a 3 Mbps CBR rate. The mean duration of the bursty phase is 1 s with a rate randomly chosen from 3 Mbps to the maximum burst rate. The burst rate changes 10 times during one bursty phase. We vary the maximum burst rate to evaluate the performance. The performance results are similar to those under CBR and Poisson patterns, which verify that our DBPR strategy is robust against such traffic bursts.

C. Performance With Varying Buffer Size

For TCP flows, the typical value of the buffer size is much smaller than the link capacity times the minimum RTT, as to achieve high link utilization and low delay [43]. In our experiments, the packet travel distance of the flows is around 9,000 KM, the total number of hops is around 6, the link capacity is 5 Mbps, and the size of all the packets is 512 B. According to the aforementioned buffer setting rule, the typical buffer size in our simulated system should be much smaller than $\frac{RTT \times 5\,\mathrm{Mbps}}{512\,\mathrm{B}} \approx 85$ packets, where $RTT = (\frac{9000\,\mathrm{KM}}{3\times10^8\,\mathrm{m/s}} + \frac{512\,\mathrm{B}}{5\,\mathrm{Mbps}} \times 6) \times 2$ seconds. To evaluate the impact of buffer size on the performance of DBPR, we vary the buffer size of all the nodes in the network from 30 to 110, and plot the performance in Fig. 11. The CBR flow rate is 3.5 Mbps.

With the increase of the buffer size, our scheme DBPR shows a slow growth in data delivery ratio, which is shown in Fig. 11(a). When the buffer size is small, DBPR shows the highest data delivery ratio as it adjusts the traffic overhead balance efficiently. We can also see that DBPR shows a similar trend with SBPR, but DBPR has a higher data delivery ratio. It indicates that both back-pressure based routing approaches can achieve a high data delivery ratio with a limited buffer size and our scheme DBPR performs better. Especially, when the buffer size is 30, DBPR can improve 50% of the data delivery ratio than LCRA. LCRA is sensitive to buffer size because it postpones transmitting data when the next hop is congested, which makes the buffer insufficient and causes many incoming data dropped. When the

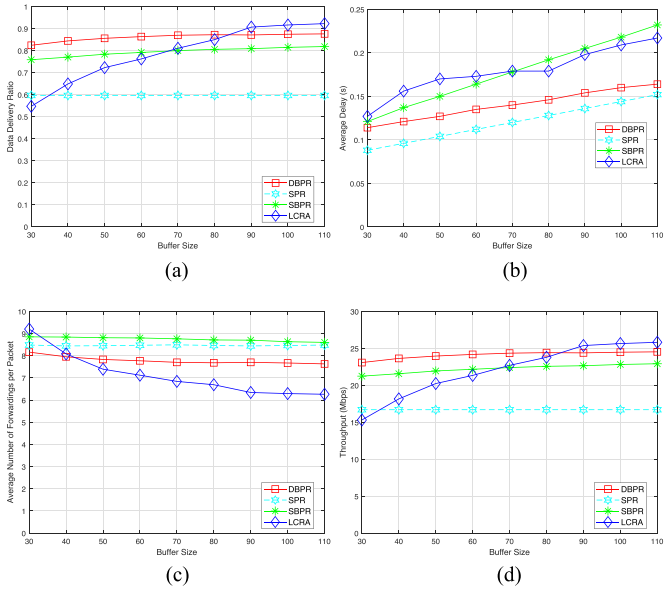


Fig. 11. The Performance of DBPR with Different Buffer Sizes. (a) Data Delivery Ratio. (b) Average Delay. (c) Average Number of Forwardings per Packet. (d) Throughput.

buffer size is large, e.g., around 90 packets, LCRA is slightly better than DBPR as it selects deterministic shortest routes to the destinations for Iridium-based satellite networks and uses neighbors' congestion information. However, this is rarely the case as the typical buffer size in such systems is much smaller than 85 packets. SPR always shows the lowest data delivery ratio among these four routing strategies as it never considers the traffic balance. The traffic is extremely congested in SPR and it shows a low data delivery ratio when the buffer size increases from 30 to 110.

The performance on the average delay is shown in Fig. 11(b). With the increase of the buffer size, the average delay of all four routing schemes increases. The reason is that more data can be buffered in a node, which makes the cache time longer and the total delay larger. SPR still shows the lowest average delay as it drops large packets and owns the lowest data delivery ratio. Comparing with sojourn-time based back-pressure routing SBPR, DBPR utilizes the distance delay between satellites of LEO satellite networks, which results in a lower average delay. LCRA postpones the traffic when the next hop is congested, which also shows a higher delay than DBPR.

With the increase of the buffer size, the average number of forwardings per packet is shown in Fig. 11(c). SBPR achieves the highest due to the large volume of data forwarded by the sojourn-time based back-pressure routing protocol. SPR also has high transmission cost due to its low data delivery ratio. When the buffer size is smaller than 50, our scheme DBPR achieves the lowest transmission cost. Such a small buffer size saves the storage resource on the satellites greatly. Note that the typical buffer size needs to be much smaller than 85 packets in our system. With the surplus supply of the buffer space, LCRA shows the lowest transmission cost. This is because when the buffer size increases, less data will be dropped in the cache and the data delivery ratio is thus increased in LCRA. LCRA selects the next hop between two shortest paths and thus it shows a low transmission cost. However, aiming at transmitting data to destinations with low delay and high data delivery ratio, it is meaningful that we spend a certain amount of transmission cost to meet the users' performance requirements and these transmission cost may be distributed among lightly loaded satellites. Therefore our scheme DBPR is still better than LCRA for providing high performance considering both average delay and data delivery ratio.

We further evaluate the throughput of the four routing schemes. Fig. 11(d) shows a similar trend on the data delivery ratio. Our protocol DBPR achieves the best throughput when the buffer size is less than 90, i.e., a buffer size smaller than 85 packets. DBPR can achieve steadily increasing throughput with the increase of the buffer resource. LCRA cannot adapt to the limited buffer environment and only shows a smaller improvement than our scheme DBPR when the buffer size is no less than 90 packets. SPR shows the lowest throughput and SBPR achieves a higher throughput than SPR.

From the performance results of Fig. 11, with the consideration of distance-delay-based back-pressure routing, our scheme DBPR can balance the buffer occupancy and traffic overhead efficiently in the network, which can achieve desirable data delivery ratio and low delay under a typically small buffer size.

VII. Conclusion

Considering the heavy traffic load between dense urban areas through LEO satellite networks, we proposed a novel distributed destination distance-delay based back-pressure routing protocol (DBPR) for LEO satellite networks. DBPR utilized the Euclidean distance between satellites to approximate the propagation delay to destinations. By combining the distance delay and back-pressure routing, DBPR selected lowly-congested and fast paths to the destinations dynamically to balance the traffic overhead in LEO satellite networks. Moreover, we designed a rectangular transmission region between sources and destinations to control the delay and transmission cost. Theoretical analysis verified that DBPR is an optimal-throughput routing scheme in LEO satellite networks. NS-2 based simulation results demonstrated that our proposed DBPR scheme can achieve high throughput and low delay without much transmission cost. It can balance the overloaded traffic dynamically and efficiently for LEO satellite networks, especially under limited-cache-resource environments.

In our future work, we will further explore more scenarios with different patterns of flows, e.g., all packets towards the same destination satellite as one flow, which can decrease the maintenance cost of virtual queues at each satellite node.

Appendix A
Satellite Distance Calculation

The distance between satellites changes periodically in LEO satellite networks and we can calculate it using the Cartesian coordinate system in Fig. 12. The geometric point O is the origin of coordinates. The x axis is on the equatorial plane and points to the vernal equinox point. The z axis is perpendicular to the

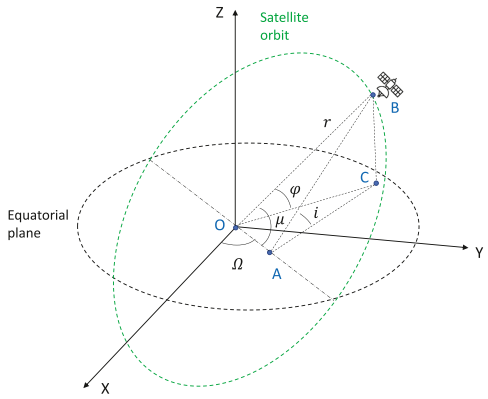


Fig. 12. The Cartesian Coordinate System for Satellites.

equatorial plane. The y axis is perpendicular to x and z axes, constituting a right-handed system.

r is the sum of earth radius and satellite orbit altitude. i is the inclination of the satellite orbit. Ω is the position of the ascending node, and the unit is degree. ω_1 is the angular velocity of a satellite rotating around the earth, and the unit is degree/second. γ is the initial phase angle of the satellite. ω_2 is the angular velocity of the earth rotation in degree/second. t is the current time.

At time t, the satellite is at position B. We first calculate the declination φ and the right ascension λ of the satellite at time t. Point C is the projection of B in the equatorial plane. Line BA is perpendicular to OA, so CA is also perpendicular to OA.

In $\triangle ABC$, $\angle ACB = \pi/2$, $\angle BAC = i$, so

$$|BC| = |AB| \times \sin i. \tag{36}$$

Set $\mu = \omega_1 t + \gamma$. In $\triangle OAB$, $\angle BAO = \pi/2$, $\angle AOB = \mu$, so

$$|AB| = |OB| \times \sin\mu. \tag{37}$$

In $\triangle OBC$, $\angle OCB = \pi/2$, $\angle BOC = \varphi$,

$$\varphi = \arcsin(|BC|/|OB|) = \arcsin(\sin i \times sin\mu). \tag{38}$$

In $\triangle AOC$, $\angle OAC = \pi/2$, so

$$\angle AOC = \arctan(|AC|/|OA|). \tag{39}$$

In $\triangle ABC$, $\angle ACB = \pi/2$, $\angle BAC = i$, so

$$|AC| = |AB| \times \cos i. \tag{40}$$

In $\triangle OAB$, $\angle BAO = \pi/2$, $\angle AOB = \mu$, so

$$|OA| = |AB|/\tan\mu. \tag{41}$$

Therefore,

$$\begin{aligned}\angle AOC &= \arctan(|AC|/|OA|) \\ &= \arctan(\cos i \times \tan\mu).\end{aligned} \tag{42}$$

Next, we calculate the coordinate of the satellite (point B) at time t.

$$x = r \times \cos\varphi \times \cos\lambda, \tag{43}$$

$$y = r \times \cos\varphi \times \sin\lambda, \tag{44}$$

$$z = r \times \sin\varphi. \tag{45}$$

We can calculate the distance L between any two satellites in the same/different orbit plane. The coordinates of the satellites j and k are (x_j, y_j, z_j) and (x_k, y_k, z_k), respectively. We have

$$L = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2}, \tag{46}$$

$$L = \sqrt{2r^2[1 - \cos\varphi_j \cos\varphi_k \cos(\lambda_j - \lambda_k) - \sin\varphi_j \sin\varphi_k]}, \tag{47}$$

$$\varphi = \arcsin(\sin i \sin\mu), \tag{48}$$

$$\lambda = \Omega + \arctan(\cos i \tan\mu) - \omega_2 t, \tag{49}$$

$$\mu = \omega_1 t + \gamma. \tag{50}$$

Now, we obtain the distance between any two satellites at time t.

Appendix B
Proof of Lemma V.1

Lemma V.1: let $\widehat{X}_n^c = \lim_{t\to\infty} \widehat{X}_n^c(t)$ and $\widehat{Y}_n^c = \lim_{t\to\infty} \widehat{Y}_n^c(t)$ be the average output and input respectively through a long-term observation. If the network is stable, the optimal transmission process satisfies

$$\lambda_n^c - \widehat{X}_n^c + \widehat{Y}_n^c \le 0. \tag{51}$$

Proof: Let $M_n^c(t)$ be the total number of packets of flow c formed at node n during time interval $[0, t]$. We divide the time into discrete points such that $M_n^c(t) = \sum_{\tau=0}^{t} m_n^c(\tau)$. $m_n^c(\tau)$ denotes the number of packets from flow c formed at node n at temporal point τ. Then the accumulated destination distance delay of $M_n^c(t)$ can be defined as

$$T\left(M_n^c(t)\right) = \sum_{\tau=0}^{t} T\left(m_n^c(\tau)\right). \tag{52}$$

Let $F_{ab}^c(t)$ be the total number of packets from flow c transmitted over link (a, b) up to time slot t. $f_{ab}^c(\tau)$ is the number of packets from flow c transmitted over link (a, b) at temporal point τ. The accumulated destination distance delay of $F_{ab}^c(t)$ is

$$T\left(F_{ab}^c(t)\right) = \sum_{\tau=0}^{t} T\left(f_{ab}^c(\tau)\right). \tag{53}$$

Considering the total number of output/input packets from flow c at node n till time slot t, we can obtain

$$T\left(\sum_{\tau=0}^{t} X_n^c(\tau)\right) = T\left(\sum_{b\in N} F_{nb}^c(t)\right), \tag{54}$$

$$T\left(\sum_{\tau=0}^{t} Y_n^c(\tau)\right) = T\left(\sum_{a\in N} F_{an}^c(t)\right). \tag{55}$$

Considering node n has the queue backlog $Q_n^c(t)$, we have

$$M_n^c(t) - Q_n^c(t) = \sum_{b\in N} F_{nb}^c(t) - \sum_{a\in N} F_{an}^c(t). \tag{56}$$

To meet the requirement of network stability in [29], [44], there exists a finite value V that makes

$$\frac{Q_n^c(t)}{t'} \leq \frac{V}{t'} < \varepsilon. \tag{57}$$

When the network is stable, for an arbitrarily small $\varepsilon > 0$ and a large time $t' > \frac{V}{\varepsilon}$, we have

$$\frac{M_n^c(t')}{t'} - \sum_{b\in N} \frac{F_{nb}^c(t')}{t'} + \sum_{a\in N} \frac{F_{an}^c(t')}{t} = \frac{Q_n^c(t')}{t'} < \varepsilon,$$

if $t' \to \infty, \varepsilon \to 0$.

Because $\lambda_n^c = \lim_{t\to\infty} \frac{M_n^c(t)}{t}$, $r_{ab}^c = \lim_{t\to\infty} \frac{F_{ab}^c(t)}{t}$, we have

$$\lambda_n^c - \sum_{b\in N} r_{nb}^c + \sum_{a\in N} r_{an}^c \leq 0. \tag{58}$$

We define $\bar{t}_n^c$ as the average destination distance delay time (slots) of flow c at node n. We know that $\bar{t}_n^c \geq 1$ is independent of time t if the network is stable.

Meanwhile, we have the following formulas for $t \to \infty$:

$$\lambda_n^c = \frac{M_n^c(t)}{t}, \tag{59}$$

$$\begin{aligned}\widehat{X}_n^c &= \frac{T\left(\sum_{\tau=0}^{t}\left(X_n^c(\tau)\right)\right)}{t} \\ &= \frac{T(\sum_{b\in N} F_{nb}^c(t))}{t} \\ &= \bar{t}_n^c \sum_{b\in N} \frac{F_{nb}^c(t)}{t},\end{aligned} \tag{60}$$

$$\begin{aligned}\widehat{Y}_n^c &= \frac{T\left(\sum_{\tau=0}^{t}\left(Y_n^c(\tau)\right)\right)}{t} \\ &= \frac{T\left(\sum_{a\in N} F_{an}^c(t)\right)}{t} \\ &= \bar{t}_n^c \sum_{a\in N} \frac{F_{an}^c(t)}{t}.\end{aligned} \tag{61}$$

When $t \to \infty$,

$$\begin{aligned}\lambda_n^c - \widehat{X}_n^c + \widehat{Y}_n^c &\leq \lambda_n^c - \bar{t}_n^c \left(\sum_{b\in N} \frac{F_{nb}^c(t)}{t} - \sum_{a\in N} \frac{F_{an}^c(t)}{t}\right) \\ &\leq \lambda_n^c - \sum_{b\in N} r_{nb}^c + \sum_{a\in N} r_{an}^c \leq 0.\end{aligned} \tag{62}$$

□

REFERENCES

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2714–2741, Oct.–Dec. 2018.

[2] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 53–87, Jan.–Mar. 2022.

[3] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.

[4] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband LEO satellite communications: Architectures and key technologies," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 55–61, Apr. 2019.

[5] W. Chien, C. Lai, M. S. Hossain, and G. Muhammad, "Heterogeneous space and terrestrial integrated networks for IoT: Architecture and challenges," *IEEE Netw.*, vol. 33, no. 1, pp. 15–21, Jan./Feb. 2019.

[6] K. Guo et al., "Performance analysis of hybrid satellite-terrestrial cooperative networks with relay selection," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9053–9067, Aug. 2020.

[7] C. E. Fossa, R. A. Raines, G. H. Gunsch, and M. A. Temple, "An overview of the IRIDIUM(R) low earth orbit (LEO) satellite system," in *Proc. IEEE Nat. Aerosp. Electron. Conf.Celebrating 50 Years (Cat. No.98CH36185)*, 1998, pp. 152–159.

[8] M. Handley, "Delay is not an option: Low latency routing in space," in *Proc. 17th ACM Workshop Hot Top. Netw.*, 2018, pp. 85–91.

[9] D. Bhattacherjee et al., "Gearing up for the 21st century space race," in *Proc. 17th ACM Workshop Hot Top Netw.*, 2018, pp. 113–119.

[10] I. D. Portillo, B. Cameron, and E. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta Astronautica*, 2018, vol. 159, pp. 123–135.

[11] J. Hu, L. Cai, C. Zhao, and J. Pan, "Directed percolation routing for ultra-reliable and low-latency services in low earth orbit (LEO) satellite networks," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC2020-Fall), New Netw. Architecture Powering Internet-of-Things Workshop*, 2020, pp. 1–6.

[12] N. Kato et al., "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.

[13] H. Huang, S. Guo, W. Liang, K. Wang, and A. Y. Zomaya, "Green data-collection from Geo-distributed IoT networks through low-earth-orbit satellites," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 3, pp. 806–816, Sep. 2019.

[14] J. Guo, D. Rincn, S. Sallent, L. Yang, X. Chen, and X. Chen, "Gateway placement optimization in LEO satellite networks based on traffic estimation," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3860–3876, Apr. 2021.

[15] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-dense LEO: Integration of satellite access networks into 5G and beyond," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 62–69, Apr. 2019.

[16] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "Joint HAP access and LEO satellite backhaul in 6G: Matching game based approaches," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1147–1159, Apr. 2021.

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

[18] W. Wang, M. Dong, K. Ota, J. Wu, J. Li, and G. Li, "CDLB: A cross-domain load balancing mechanism for software defined networks in cloud data centre," *Int. J. Comput. Sci. Eng.*, vol. 18, no. 1, pp. 44–53, 2019.

[19] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Congestion-aware scheduling for software-defined SAG networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2861–2871, Oct.–Dec. 2021.

[20] L. Chen, F. Tang, and X. Li, "Mobility- and load-adaptive controller placement and assignment in LEO satellite networks," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[21] T. Taleb, D. Mashimo, A. Jamalipour, N. Kato, and Y. Nemoto, "Explicit load balancing technique for NGEO satellite IP networks with on-board processing capabilities," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 281–293, Feb. 2009.

[22] Z. Zhang, C. Jiang, S. Guo, Y. Qian, and Y. Ren, "Temporal centrality-balanced traffic management for space satellite networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4427–4439, May 2018.

[23] H. Nishiyama, D. Kudoh, N. Kato, and N. Kadowaki, "Load balancing and QoS provisioning based on congestion prediction for GEO/LEO hybrid satellite networks," *Proc. IEEE*, vol. 99, no. 11, pp. 1998–2007, Nov. 2011.

[24] F. Tang, H. Zhang, and L. T. Yang, "Multipath cooperative routing with efficient acknowledgement for LEO satellite networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 179–192, Jan. 2019.

[25] X. Li, F. Tang, L. Chen, and J. Li, "A state-aware and load-balanced routing model for LEO satellite networks," in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–6.

[26] Z. Liu, J. Li, Y. Wang, X. Li, and S. Chen, "HGL: A hybrid global-local load balancing routing scheme for the Internet of Things through satellite networks," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 3, pp. 1–16, 2017.

[27] X. Liu, X. Yan, Z. Jiang, C. Li, and Y. Yang, "A low-complexity routing algorithm based on load balancing for LEO satellite networks," in *Proc. IEEE 82nd Veh. Technol. Conf.*, 2015, pp. 1–5.

[28] Z. Jiao, B. Zhang, C. Li, and H. T. Mouftah, "Backpressure-based routing and scheduling protocols for wireless multihop networks: A survey," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 102–110, Feb. 2016.

[29] L. Hai, Q. Gao, J. Wang, H. Zhuang, and P. Wang, "Delay-optimal backpressure routing algorithm for multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2617–2630, Mar. 2018.

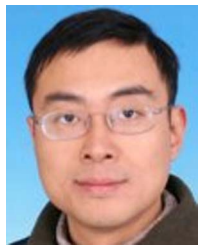
[30] A. Rai, C. Li, G. Paschos, and E. Modiano, "Loop-free backpressure routing using link-reversal algorithms," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2988–3002, Oct. 2017.
[31] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 841–854, Jun. 2011.
[32] X. Lan, Y. Chen, and L. Cai, "Throughput-optimal H-QMW scheduling for hybrid wireless networks with persistent and dynamic flows," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1182–1195, Feb. 2020.
[33] L. Wood, A. Clerget, I. Andrikopoulos, G. Pavlou, and W. Dabbous, "IP routing issues in satellite constellation networks," *Int. J. Satell. Commun.*, vol. 19, pp. 69–92, 2001.
[34] R. Kucukates and C. Ersoy, "Minimum flow maximum residual routing in LEO satellite networks using routing set," *Wireless Netw.*, vol. 14, no. 4, pp. 501–517, 2008.
[35] J. Chen, L. Liu, and X. Hu, "Towards a throughput-optimal routing algorithm for data collection on satellite networks," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 7, pp. 1–15, 2016.
[36] E. Ekici, I. F. Akyildiz, and M. D. Bender, "A multicast routing algorithm for LEO satellite IP networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 183–192, Apr. 2002.
[37] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
[38] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
[39] Y. Chen, X. Wang, and L. Cai, "On achieving fair and throughput-optimal scheduling for TCP flows in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 12, pp. 7996–8008, Dec. 2016.
[40] Y. Chen, X. Wang, and L. Cai, "HOL delay based scheduling in wireless networks with flow-level dynamics," in *Proc. IEEE Glob. Commun. Conf.*, 2014, pp. 4898–4903.
[41] M. Alresaini, K. Wright, B. Krishnamachari, and M. J. Neely, "Back-pressure delay enhancement for encounter-based mobile networks while sustaining throughput optimality," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1196–1208, Apr. 2016.
[42] Y. Cui, E. M. Yeh, and R. Liu, "Enhancing the delay performance of dynamic backpressure algorithms," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 954–967, Apr. 2016.
[43] Y. Ganjali and N. Mckeown, "Update on buffer sizing in internet routers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 67–70, 2006.
[44] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Cambridge, MA, USA: Massachusetts Inst. Technol., 2003.

Xia Deng (Member, IEEE) received the B.S. degree in electronic information engineering, and the M.S. and Ph.D. degrees in computer application technology from Central South University, Changsha, China, in 2003, 2006, and 2015, respectively. In 2006, she joined the School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China, where she is currently an Associate Professor. She was the recipient of the Best Paper Award at IEEE ICC 2013. Her research interests include LEO satellite networks, edge computing, mobile social networks and protocol design, and optimization in mobile networks.

Le Chang (Member, IEEE) received the B.S. and M.S. degrees in computer science from Central South University, Changsha, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada, in 2013. From 2013 to 2017, he was a Research Engineer with the Central Research Institute with Huawei Technologies Co., Ltd. Since 2017, he has been an Assistant Professor with the School of Automation, Guangdong University of Technology, Guangzhou, China. His research interests include various distributed systems and the design, and optimization of network protocols.

Shouyuan Zeng received the B.S. degree in software engineering from Chang'an University, Xi'an, China, in 2018, and the M.E. degree in computer technology, Guangzhou University, Guangzhou, China, in 2022. His research interests include LEO satellite networks, mobile computing, and routing protocol design and analysis.

Lin Cai (Fellow, IEEE) received the M.A.Sc. and Ph.D. degrees (awarded Outstanding Achievement in Graduate Studies) in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, and she is currently a Professor. She is an NSERC E.W.R. Steacie Memorial Fellow, an Engineering Institute of Canada Fellow. In 2020, she was elected as a Member of the Royal Society of Canada's College of New Scholars, Artists and Scientists. She was also elected as a 2020 Star in Computer Networking and Communications by N2Women. Her research interests include communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and the Internet of Things. She was the recipient of the NSERC Discovery Accelerator Supplement Grants in 2010 and 2015, respectively, and the Best Paper awards of IEEE ICC 2008 and IEEE WCNC 2011. She has co-founded and chaired the IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She was elected to serve the IEEE Vehicular Technology Society Board of Governors, 2019–2024. She was an Associate Editor-in-Chief (AEiC) for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a Member of the Steering Committee of the IEEE TRANSACTIONS ON BIG DATA and IEEE TRANSACTIONS ON CLOUD COMPUTING, an Associate Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COMMUNICATIONS, *EURASIP Journal on Wireless Communications and Networking*, *International Journal of Sensor Networks*, and *Journal of Communications and Networks*, and as the Distinguished Lecturer of the IEEE VTS Society and the IEEE ComSoc Society. She was a TPC Co-Chair of the IEEE VTC2020-Fall, and a TPC Symposium Co-Chair of the IEEE Globecom'10 and Globecom'13. She is a registered Professional Engineer with British Columbia, Vancouver, BC, Canada.



Jianping Pan (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from Southeast University, Nanjing, China, and he did his Postdoctoral research with the University of Waterloo, Waterloo, ON, Canada. He is currently a Professor of computer science with the University of Victoria, Victoria, BC, Canada, British Columbia, Vancouver, BC, Canada. He also worked at Fujitsu Labs and NTT Labs. His area of specialization is computer networks and distributed systems, and his research interests include protocols for advanced networking, performance analysis of networked systems, and applied network security. He was the recipient of the IEICE Best Paper Award in 2009, the Telecommunications Advancement Foundation's Telesys Award in 2010, the WCSP 2011 Best Paper Award, the IEEE Globecom 2011 Best Paper Award, the JSPS Invitation Fellowship in 2012, IEEE ICC 2013 Best Paper Award, and the NSERC DAS Award in 2016, and was the Technical Program committees of major computer communications and networking conferences including IEEE INFOCOM, ICC, Globecom, WCNC and CCNC. He was the Ad Hoc and Sensor Networking Symposium Co-Chair of IEEE Globecom 2012 and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is a senior member of the ACM.