FALCON: Towards Fast and Scalable Data Delivery for Emerging Earth Observation Constellations

Mingyang Lyu[‡], Qian Wu[‡*], Zeqi Lai[‡*✉], Hewu Li[‡*], Yuanjie Li[‡*], Jun Liu[‡*]
[‡]*Institute for Network Sciences and Cyberspace*, Tsinghua University, [*]*Zhongguancun Laboratory*, Beijing, China

IEEE INFOCOM 2023 - IEEE Conference on Computer Communications | 979-8-3503-3414-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOM53939.2023.10228910

***Abstract*—Exploiting a constellation of small satellites to realize continuous earth observations (EO) is gaining popularity. Large-volume EO data acquired from space needs to be transferred to the ground. However, existing EO delivery approaches are either: (a) *efficiency-limited*, suffering from long delivery completion time due to the intermittent ground-space communication, or (b) *scalability-limited* since they fail to support concurrent delivery for multiple satellites in an EO constellation.**

To make big data delivery for emerging EO constellations *fast* and *scalable*, we propose FALCON, a multi-path EO delivery framework that wisely exploits diverse paths in broadband constellations to collaboratively deliver EO data effectively. In particular, we formulate the constellation-wide EO data multi-path download (CEOMD) problem, which aims at minimizing the delivery completion time of requested data for all EO sources. We prove the hardness of solving CEOMD, and further present a heuristic multipath routing and bandwidth allocation mechanism to tackle the technical challenges caused by time-varying satellite dynamics and flow contention, and solve the CEOMD problem efficiently. Evaluation results based on public orbital data of real EO constellations show that as compared to other state-of-the-art approaches, FALCON can reduce at least 51% delivery completion time for various data requests in large EO constellations.

I. INTRODUCTION

Thanks to the recent technique breakthrough in the sensing and aerospace industry, earth observation (EO) technologies are evolving rapidly in the past decade. It is estimated that the revenues of EO data and services are forecast to double from roughly €2.8 billion to over €5.5 billion over the next decade [44].

Two critical trends can be observed from the recent evolution of the EO ecosystems. First, emerging EO satellites are equipped with multiple high-resolution sensors to capture EO data from space [12], [13], [32] for various missions, *e.g.,* fine-granularity environment monitoring and disaster prediction. Second, many EO service providers tend to leverage a large number of EO satellites (*i.e.,* a constellation) to cooperatively execute EO missions. The revisit time can be significantly reduced by cooperatively using EO satellites in a constellation for observation. This is because EO satellites typically work in low earth orbit (LEO) close to the earth surface and move at a high velocity, and the revisit time of a single EO satellite could be very long (*e.g.,* several hours or days) [11], [50].

Taking the above two trends together, emerging EO constellations are generating a large volume of data every day [21]. In urgent cases like disaster response, EO data acquired in space needs to be downloaded to the ground mission center as fast as possible. Therefore, optimizing the data delivery process for EO constellations is critical for the EO industry.

✉ Zeqi Lai is the corresponding author.

Many prior works have studied the methods for delivering data from EO satellites. At a high level, existing methods can be divided into three categories: (a) downloading via ground station networks [48]; (b) downloading via geostationary (GEO) relays [40]; and (c) downloading via LEO satellite routes [25]. Specifically, the first category exploits a collection of distributed ground stations to download EO data when the satellite carrying data moves into the transmission range of a certain ground station. However, this method fails to sustain long-duration data download and suffers from high delivery completion time due to the limited deployment of available ground stations and intermittent ground-space communication. The second category leverages geostationary (GEO) satellite relays to download EO data to the ground via persistent and geostationary forwarding paths. While robust, this method has very limited scalability since it can not support large-scale EO constellations due to the limited amount of transmission links in GEO relays. Although the third category inter-connects satellites by inter-satellite links (ISL) to establish high-speed download paths from the EO satellite to its ground destinations, this method is not fast and scalable enough due to: (a) the path contention when the number of EO sources increases; and (b) frequent network disruptions caused by the high-velocity movement of satellites. Therefore, we ask a pragmatic question: *is there a viable path to deliver big EO data collected from emerging EO constellations to the ground in a fast and scalable manner?*

In this paper, we affirmatively answer the question above by presenting FALCON, a multipath EO delivery framework. Falcon mainly adopts two key ideas to enable fast and scalable big data delivery for EO constellations: (a) *download big EO data in an on-demand way.* That is, it is unnecessary to wait for all the satellites to finish downloading all the data they've collected, we only download the requested EO data that is related to the area of interest. This can greatly reduce the amount of data we need to transfer. (b) *exploit the high multipath diversity in broadband mega-constellation networks to concurrently establish multiple delivery paths.* In this way, the download throughput can be improved while enhancing the ability to resist the impact of frequent handover between satellites and ground stations.

However, the dynamic fluctuations in the core infrastructure of mega-constellation networks involve new challenges on applying multipath download for EO tasks: the time-varying topology fluctuations accordingly result in route and traffic fluctuations. Moreover, carrying a large number of EO delivery flows can lead to excessive congestion with shared bottleneck links. Hence, multiple routes should be dynamically and wisely re-calculated in a time-varying manner, and flows should be adaptively re-scheduled in each route, to avoid link congestion and obtain transmission efficiency.

Our mechanism addresses the above challenges in two steps. First, we model the dynamic and hybrid constellation topology, as well as its time-varying network capacity and availability. Combining EO constellations and broadband mega-constellations, we formulate the Constellation-wide EO data Multipath Download (CEOMD) problem, which aims at minimizing the time when all the sources complete the transmission of requested data. We also demonstrate the hardness of solving CEOMD problem under representative EO scenarios.

Second, to solve the CEOMD problem efficiently, we propose a Heuristic Multipath Routing and Bandwidth Allocation (HMRBA) algorithm. The key principle behind our algorithm is to select paths with low inter-path link overlap to avoid severe network congestion in a greedy order where the source with more data comes first, and preferably choose those paths that have longer stable time to transmit data. When allocating bandwidth for each path, we proportionally distribute the bandwidth resources according to the data volume of each source avoiding the ones with much data from costing too much time to transfer.

We evaluate FALCON by simulations based on real and public constellation information. Extensive evaluation results demonstrate that by dynamically constructing multiple download paths and judiciously allocating EO traffic upon them, FALCON can reduce at least 51% of the delivery completion time when serving various data requests in large EO constellations, as compared to other state-of-the-art approaches,.

In conclusion, this paper makes three contributions.

- **(a)** Exposing the importance and challenges of optimizing data delivery for EO constellations, with a formulation of the Constellation-wide EO data Multipath Download (CEOMD) problem, which is NP-hard.
- **(b)** Presenting a novel framework called FALCON to achieve fast and scalable constellation-wide EO data download, by adopting a Heuristic Multipath Routing and Bandwidth Allocation (HMRBA) algorithm.
- **(c)** Demonstrating the effectiveness of FALCON by extensive simulations based on real and public information obtained from the satellite ecosystem.

II. MOTIVATION AND RELATED WORK

A. Research Background: Observing the Earth via Satellites

Various earth observation missions. The earth observation (EO) ecosystem is continually evolving, toward seamless integration of new technologies, sensing modalities, and unconventional data sources. According to a recent report [45], nearly 45% of existing low earth orbit (LEO) satellites in space are launched for various EO missions, such as forest observation, weather forecasting, agriculture monitoring, crisis management and maritime surveillance, *etc.* Essential information on global areas collected by EO satellites enables us to monitor

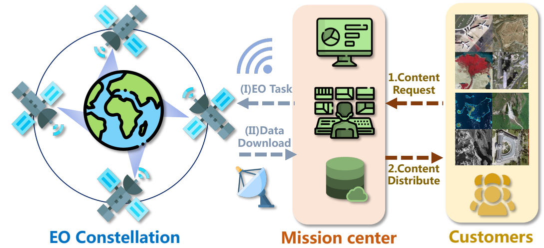

Fig. 1. The service model of existing EO ecosystems.

and protect our environment, manage our resources, respond to global disasters and enable sustainable development.

EO service model. Fig. 1 briefly plots the service model of existing EO ecosystems. At a high level, there are two major interactions. First, an EO service provider interacts with satellites via ground stations to assign EO tasks and collect data. In particular, an EO service provider owns and operates a number of EO satellites to perform EO tasks, *e.g.,* acquiring information for a specific region of interest. To download data from space, existing commercial EO systems follow a "store first, download later" model, where space data is first acquired and stored in the satellite storage. All data will be downloaded to the ground for further processing and storage, via ground-satellite links (*e.g.,* [47]), or satellite relays (*e.g.,* [14]). Second, customers who require EO data for their own applications (*e.g.,* remote monitoring) interact with the service provider via terrestrial Internet. Note that if the required contents have already been saved in the storage, the service provider sends them directly back to the customers. Otherwise, the service provider has to establish a new EO task to collect data, and then distribute content to the customers.

B. New Trends and Requirements Facing the EO Industry

With the rapid technical evolution in aerospace and remote sensing technologies, in recent years we have witnessed two critical trends in the EO industry.

T(1): from monolithic satellite to satellite constellations. Due to the high dynamics of LEO satellites and earth rotation, it is difficult for a single satellite to achieve high *temporal resolution* and maintain continuous observation in an EO mission. In particular, it may take hours to days for a satellite to revisit a specific region of interest upon the earth's surface. Therefore, recent EO service providers leverage *a constellation of EO satellites* to *cooperatively* acquire data and reduce the revisiting time of a specific area of interest (AoI). For example, Planet [34] has launched and deployed 452 satellites consisting three constellations: PlanetScope [2], RapidEye [3], SkySat [4] to collaboratively perform EO missions. They can capture earth's activities from multiple perspectives and dimensions with revisiting time less than one day.

T(2): from low-quality to high-quality space sensors. In the early 1980s, the spatial resolution of EO satellites was around 30 meters as on LandSat-4 [46]. With the development of techniques, as low as 30 centimeters of spatial resolution is available in state-of-the-art EO satellites. Even CubeSats in PlanetScope can achieve a spatial resolution of around 3 meters [36]. The spectral resolution also improved dramatically over the past few decades, as sensors were refined and more bands became available for study. Some state-of-the-art satellite sensors can now capture pictures with more than 1000 spectral bands [39].

In parallel with the new trends above, ideally emerging EO systems are expected to satisfy two critical performance requirements *simultaneously*, as described below.

R(1): fast information delivery. It is expected that the EO data can be downlinked to the ground as soon as possible. This is especially important for time-sensitive EO tasks, *e.g.,* requiring fresh information from a wildfire or rescue scene.

R(2): scalability. As EO service providers leverage EO constellations to serve earth surveillance, it is expected that an EO system can maintain acceptable availability and performance as the number of EO sources grows up even in urgent cases.

C. Related Works

Many recent efforts have been proposed to optimize the data delivery for EO systems, which can be concluded in three aspects as described below.

Ground station networks. Many existing EO systems are using ground station networks directly to downlink the EO data, like [10]. But they mainly use a limited number of big ground stations which are high-cost to deploy and maintain, inducing high latency to deliver EO data for large EO constellations. Recent research [48] is proposed to use low-cost ground stations which are distributed all over the world to offer low latency downlink by downloading the data successively and cooperatively. This approach reduces the download time by increasing the time that a satellite can communicate with a ground station in one pass. However, since EO satellites move at high velocity, the visible window can only last for several minutes. Once an EO satellite moves out of the transmission range of a ground station, it has to interrupt the transmission process, waiting for another available ground station to continue the download process. Thus, the primary limitation of downloading data by distributed ground station networks is that: ground stations are difficult to be deployed on oceans which occupy nearly 70% of our earth surface, causing intermittent download and increasing the download completion time. Moreover, to take such an unprecedented amount of data to the ground is undoubtedly expensive and long duration.

Satellite networks. Another prevalent approach for EO data transmission is leveraging GEO satellite relay networks, such as the European Data Relay Satellite (EDRS) system [18] owned by ESA, and Tracking and Data Relay Satellite (TDRS) system [40] operated by NASA. The key idea behind this method is to use satellite relay in geostationary orbit to establish long-duration and reliable LEO-to-GEO-to-ground communication path to transfer data acquired by EO satellites. However, downloading data by GEO satellite relays is *scalability-limited*, and is difficult to support a number of satellites in the EO constellation. Specifically, only 2 user-spacecrafts can be connected to a TDRS relay [40] at the same time due to the limited on-board weight available for high-speed laser communication components. In addition, it's economically difficult to launch many GEO relays to support more LEO satellites, since the cost to manufacture and launch a GEO satellite with the laser communication components is extremely high, e.g. $544 million for one EDRS relay [18].

Recent broadband LEO mega-constellations like Starlink [41] and Kuiper [5] are gaining popularity, which consist of thousands of inter-connected satellites with laser inter-satellite links (ISLs). These mega-constellations promise to offer capacities up to 20Gbps [8], and provide broadband Internet service with lower latency [16]. A collection of recent works [16], [17], [25], [42] have proposed to leverage ISLs in mega-constellations to establish space routes consisting of ISLs and ground-satellite links (GSLs) for low-latency, high-speed data transmission. These prior efforts on LEO satellite routing suggest another viable path to download big EO data from space: exploiting multi-hop satellite routes from the EO satellite to transfer data to ground destinations. However, directly exploiting broadband constellations to download EO data can inevitably impose significant challenges for satellite systems. Continuously activating download links for high-volume EO data involves high energy consumption, which further requires to increase the size of battery or solar panel, involving big challenges on satisfying the stringent constraints on the mass, volume and cost of satellites.

In-orbit data filtering. To decrease the amount of data that should be downloaded to the ground, orbital edge computing (OEC) [9] was proposed, which exploits improved onboard computing resources together with deep learning to filter out the valuable parts from the raw EO data. However, the "valuable" parts are hard to define, for different EO missions, the valuable items differ. For forest monitoring, the pictures of the sea may be useless but not in maritime surveillance cases. Other works like [15] also propose to transfer the processing process to the satellite edges to reduce the bandwidth consumption to accelerate the transmission process by leveraging emerging commercial off-the-shelf (COTS) system-on-chip (SoC) technologies. However, other than the limitations above, it can be also time-consuming to wait for visible ground stations since they don't use the LEO satellite networks.

Summarily, prior solutions for EO delivery, are either *efficiency-limited* (like ground station networks) since they suffer from long delivery completion time due to the intermittent space-ground connectivity, or *scalability-limited* (like satellite networks), since they are unable to guarantee good delivery performance when the EO constellation size scales up. The *state quo* thus motivates us to explore a new solution to accomplish fast and scalable data delivery for EO constellations.

III. The Falcon design

We propose Falcon (**F**ast, and Sc**AL**able **C**onstellation-wide Earth **O**bservatio**N**), a multipath EO delivery framework aiming at enabling fast and scalable EO data delivery for EO constellations.

A. Falcon *Overview*

To avoid a long time to wait for visible ground stations as in L2D2, we leverage space routes over inter-satellite links (ISLs) and ground-satellite links (GSLs) to achieve fast and scalable EO data delivery for constellations. However, it involves two

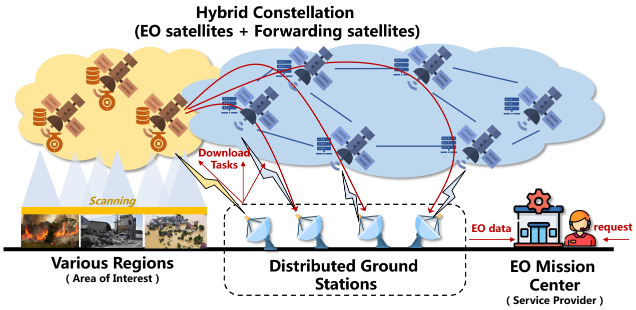

Fig. 2. FALCON system architecture.

challenges in practice. First, EO and broadband satellites are highly dynamic. Routes from the source to the terrestrial destination are likely to be interrupted by inevitable satellite-ground handoffs. Thus, it is difficult to make the download process stable and guarantee that on-demand requests can be satisfied on time. Second, as the number of sources with request data in EO constellations expands, it is challenging to properly schedule a large number of download flows from EO satellites to the terrestrial destination under a highly-dynamic network topology to avoid bandwidth contention.

To make the data transmission robust and satisfy EO requests as fast as possible, FALCON explores the *path diversity* of mega-constellation networks, and **exploits a dynamic multi-path download mechanism to aggregate bandwidth and attain stable transmission.** Emerging mega-constellations like Starlink and Kuiper follow the *Walker Delta* constellation design [49], where the ascending nodes of the planes are distributed over the full range of 360 degrees, and satellites are evenly spaced in their orbits. Therefore, a mega-constellation network is essentially an evenly distributed network with high *path diversity* [26]. In other words, there exists a number of shortest paths between the same source and destination in a mega-constellation. Exploiting multi-path delivery enables two advantages for EO download: (1) the source EO satellite can access more bandwidth by using multiple space paths simultaneously and attain faster data download; (2) if multiple download paths exist, the resilience to the handover events can be improved since EO traffic can quickly switch to an alternate path when a satellite-ground interruption occurs.

To handle the circumstances where there are tens or even hundreds of sources all with requested data to transmit, we propose a **heuristic EO multipath routing and bandwidth allocation algorithm** to judiciously schedule a large number of download flows in the dynamic mega-constellation network, avoiding link congestion and sustaining high throughput for each EO tasks at scale. Notice that, in our framework, instead of waiting for all data collected from space to finish downloading, we only download the requested data of AoIs.

B. System Architecture and Use Phases

Fig. 2 plots the high-level system architecture of FALCON. Collectively, FALCON contains an EO mission center which is operated by the EO service provider, and a hybrid constellation integrating EO and broadband satellites to cooperatively perform the download tasks. The mission center processes the user requests, like the government needs information for disaster emergency response, which contains the AoI locations and the request time duration, assigns download tasks to the EO constellation, and delivers required contents back to the users. EO satellites in the hybrid constellation are equipped with high-volume ISLs.

At runtime, EO satellites continuously gather information from various regions around the world and save EO data in their own storage. Every image is tagged with the location information and the timestamp when the source takes the photo. When the request for information on a specific area is received by the mission center, the mission center calculates which EO satellites contain the required data by checking which has passed over the AoI in the request duration. This can be achieved by reverse extrapolation of the satellite's trajectory using the public two-line-element (TLE) information. After that, the mission center assigns download tasks to all EO satellites carrying the required information via ground station networks or LEO networks, and invokes the multipath routing and bandwidth allocation mechanism to establish high-throughput download sessions while avoiding severe link congestions. In practice, the routing mechanism can be achieved by using source routing based on the routing protocols in the network and the bandwidth allocation part can be implemented by Traffic Control (TC) tools. Finally, when all related data have been gathered by the mission center, it combines and sends them back to the user. In the following sections, we introduce the details of our multipath routing and bandwidth allocation algorithm.

IV. MODELS AND PROBLEM FORMULATION

A. Network Models.

Dynamic network graph. We use a dynamic graph to characterize the high dynamics of the hybrid constellation-constellation-ground networks. Although the connections between LEO satellites and ground stations typically change within minutes, as do EO and broadband satellites, the entire constellation topology can be seen as constant over short periods of time. Thus, assume time is slotted, and the network topology in slot $t \in \mathcal{T}\{1, 2, ..., T\}$ is presented by a graph $\mathcal{G}_t(\mathcal{V}, \mathcal{E}(t))$, where T is what we called mission period which is assumed to be large enough that every source can finish the data transmission. Here, the time duration of each slot is set to be 1 second. The vertex set $\mathcal{V}$ contains the EO mission center (*i.e.,* the terrestrial destination), available ground stations and all satellites (*i.e.,* both EO satellites that acquire original data in space, and forwarding satellites that forward EO data from the source to terrestrial destination). Specifically, we assume there are n EO satellites $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ indexed by i, which have data to transmit for a certain EO data request. The corresponding data volume that has to be delivered to the destination is defined as $\mathcal{D} = \{d_1, d_2, ..., d_n\}$. The edge set $\mathcal{E}(t)$ describes the inter-vertex connectivity in slot t. An edge $e(a, b) \in \mathcal{E}(t), a, b \in \mathcal{V}$ indicates that there is an available link between node a and b in slot t. In particular, (a, b) could be an inter-satellite link (ISL), or a ground-satellite link (GSL). Due to the high dynamics of LEO satellites, $\mathcal{E}(t)$ may change in different time slots. Two vertexes in the graph can establish a link if they are visible to each other which we can pre-calculate based on the nodes' predictable location. In particular, we follow the well-known +Grid connectivity pattern [6], [16], [20] to interconnect forwarding satellites. Each forwarding satellite connects to its four adjacent satellites: two in the same orbit, and two in the left/right adjacent orbit. As for GSL, as long as the EO satellites or ground stations are under the transmission coverage of forwarding satellites, we assume there is a link between them. Furthermore, $\mathcal{H}_{ab}^{t}$ is used to denote the capacity of edge $(a,b) \in \mathcal{E}(t)$.

Multi-path and subflows. Suppose that every source has ψ interfaces. Then there could be at most ψ subflows that a source can establish with the destination concurrently. The subflow indexed by j of the ith source at time slot t is represented as $f_{ij}(t)$. $r_{ij}(t)$ is the corresponding data rate of subflow $f_{ij}(t)$. As for the path that $f_{ij}(t)$ passes through, we define a binary variable $x_{ij}^{ab}(t)$ as the path indicator which is presented as follows:

$$x_{ij}^{ab} = \begin{cases} 1, & if\ f_{ij}(t)\ passes\ through\ edge\ (a,b) \\ 0, & otherwise \end{cases}. \tag{1}$$

B. Problem Formulation.

Collectively, our primary goal is to download all the relevant EO data of certain AoI in the request duration from EO constellations as fast as possible. Formally, the Constellation-wide EO data Multipath Download **(CEOMD)** problem can be formulated as follows.

Inputs: (1) network topology $\mathcal{G}_t(\mathcal{V}, \mathcal{E}(t))$ in each slot in the mission period T; (2) link capacity matrix $\{\mathcal{H}_{ab}(t) | \forall (a,b) \in \mathcal{E}(t)\}$; (3) sources with requested data $\mathcal{S} = \{s_1, s_2, ..., s_n\} \subset \mathcal{V}$; (4) The total data volume of each source that needs to transfer $\mathcal{D} = \{d_1, d_2, ..., d_n\}$; (5) destination $dst \in \mathcal{V}$.

Outputs: (1) path indicator for every subflow of every source at each time slot $x_{ij}^{ab}(t), \forall 1 \le i \le n, \forall 1 \le j \le \psi, \forall t \in \mathcal{T}$; (3) subflow data rate allocation of each flow $r_{ij}(t)$, $\forall 1 \le i \le n, \forall 1 \le j \le \psi, \forall t \in \mathcal{T}$.

Objective:

$$minimize \max_{i=1,...,n} \tau_i \tag{2}$$

Subject to:

$$\tau_i = \min\{t' | \sum_{t=1}^{t'} \sum_{j=1}^{\psi} r_{ij}(t) = d_i, t \in \mathcal{T}\}, \forall i \in [1,n] \cap \mathbb{Z} \tag{3}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{\psi} x_{ij}^{ab}(t) * r_{ij}(t) \le H_{ab}^{t}, \forall (a,b) \in \mathcal{E}(t), \forall t \in \mathcal{T} \tag{4}$$

$$\begin{gathered} \sum_{b:(a,b)\in E_t} x_{ij}^{ab}(t) - \sum_{b:(b,a)\in E_t} x_{ij}^{ba}(t) = \pi(a,i,j,t), \\ \forall i \in [1,n] \cap \mathbb{Z}, \forall j \in [1,\psi] \cap \mathbb{Z} \end{gathered} \tag{5}$$

$$r_{ij}(t) \ge 0, \forall i \in [1,n] \cap \mathbb{Z}, \forall i \in [1,\psi] \cap \mathbb{Z}, \forall t \in \mathcal{T} \tag{6}$$

where:

$$\pi(a,i,j,t) = \begin{cases} 1, & if\ a = s_i \\ -1, & if\ a = dst \\ 0, & otherwise \end{cases}. \tag{7}$$

Objective (2) is to minimize the maximum transmission completion time of each source, which we call the overall delivery completion time. τ_i is defined in constraint (3) to represent the completion time of source s_i, which is the minimum time slot when there is no data that needs to be transmitted. Constraint (4) indicates that all flows passing the same link should satisfy that the sum of their data rate doesn't exceed the link capacity. Constraint (5) ensures flow conservation, *e.g.,* for any subflow $f_{ij}(t)$ and any intermediate node a, the number of links through which data is transferred into node a should be equal to the number of links through which data is sent out from node a. It also indicates that for any sublfow it uses only one link out of the source node and one link into the destination. The final constraint limits the subflow data rate to be nonnegative values.

To make the formulation clearer, we first introduce a new binary integer indicator $y_i(t)$ to represent whether s_i finishes the data transmission at time slot t (0 for finished, 1 for not). Then, we get the constraints of $y_i(t)$ as follows:

$$y_i(t) \ge y_i(t+1), y_i(t) \in \{0,1\}, \forall t \in \mathcal{T}, \tag{8}$$

since $y_i(t)$ is a non-increasing variable. Then $\tau_i = \sum_{t=1}^{T} y_i(t)$, we merge it with the objective function (2) and get the final problem formulation as follows:

$$minimize \max_{i=1,...,n} \sum_{t=1}^{T} y_i(t) \tag{9}$$

Subject to:

$$\sum_{j=1}^{\psi} r_{ij}(t) > y_i(t) - y_i(t+1) - 1, \forall i \in [1,n] \cap \mathbb{Z}, \forall t \in \mathcal{T} \tag{10}$$

$$y_i(t) \ge y_i(t+1), 0 \le y_i(t) \le 1, \forall t \in \mathcal{T} \tag{11}$$

$$\sum_{t=1}^{T} \sum_{j=1}^{\psi} r_{ij}(t) = d_i, \tag{12}$$

and constraints (4)(5)(6).

The CEOMD problem is a **mixed non-linear problem** with infinite (although countable) both integer and non-integer variables which is hard to analyze its complexity. However, we can simplify this question and prove it as an NP-hard problem.

Theorem 1. *The CEOMD problem is NP-hard.*

Proof. We reduce this CEOMD problem to the Coflow Routing and Scheduling Optimization (CRSO) problem in data centers claimed in [51], which has been proved to be an NP-hard problem. The CRSO problem is to route and schedule the coflow in data centers to minimize the data transmission completion time. We first simplify CEOMD by setting the network topology to be static, then the $\mathcal{G}(\mathcal{V}, \mathcal{E})$ can be mapped to the topology of data center networks. After that, we limit the number of subflow of each source to one so that subflows of each source can be regarded as in a coflow. Further, if the paths for each flow are pre-defined, this problem is totally a CRSO problem. Hence the simplified CEOMD problem is NP-hard, as well as the original CEOMD problem.

Algorithm 1: Paths and alive time calculation

Input: topology graph $\mathcal{G}_t(\mathcal{V}, \mathcal{E}(t))$ at every time slot t, Source list $\mathcal{S} = \{s_1, s_2, ..., s_n\}$, ground stations $GS = \{gs_1, gs_2, ...\}$, destination Dst, current time slot t_0

Output: candidate path of each source P_{t_0}, and their corresponding alive time

Init $P_{it_0} = \varnothing, \forall i = 1, ..., n$;
for $i \leftarrow 1$ *to* n **do**
    for $gs \in GS$ **do**
        $tmp\_P = cal\_all\_shortest\_path(\mathcal{G}_t, s_i, gs)$;
        for $p \in tmp\_P$ **do**
            $p \leftarrow p \cup (gs, dst)$;
            $P_{it_0} \leftarrow P_{it_0} \cup p$;
            $\mathbf{t}_p \leftarrow \min\{t | e \notin \mathcal{E}(t), \forall e \in p\}$
$P_{t_0} \leftarrow \{P_{1t_0}, P_{2t_0}, ..., P_{nt_0}\}$;
return $P_{t_0}, \mathbf{t}_p, \forall p \in P_{t_0}$

Algorithm 2: Heuristic Multipath Routing and Bandwidth Allocation (HMRBA) algorithm

Input: topology graph $\mathcal{G}_t(\mathcal{V}, \mathcal{E}(t))$ at every time slot t, Source list $\mathcal{S} = \{s_1, s_2, ..., s_n\}$, data volume $\mathcal{D} = \{d_1, d_2, ..., d_n\}$, destination Dst

Output: path indicator $x_{ij}^{ab}(t)$, subflow data rate $r_{ij}(t)$ at every time slot t

Init current using path set at $t = 0$: $\mathbb{P}_0 = \{P_{i0}^* = \varnothing | \forall i\}$
Init the next path set change time $\mathbf{t} = 0$
for $t \in \mathcal{T}$ **do**
    if $\mathbf{t} > t$ **then**
        $x_{ij}^{ab}(t) \leftarrow x_{ij}^{ab}(t-1), \forall i, j$
        if *no source complete* **then**
            $r_{ij}(t) \leftarrow r_{ij}(t-1)$; continue;
    else
        //path selection
        delete invalid path from $\mathbb{P}_{t-1}$ get $\mathbb{P}_t$;
        sort S in the descending order of data volume;
        for $i \leftarrow 1$ *to* n **do**
            while $|P_{it}^*| < \psi$ **do**
                get candidate paths P_{it};
                $p^* = arg\min\{M_p | p \in P_{it}\}$;
                $P_{it}^* \leftarrow P_{it}^* \cup p^*$;
    update $x_{ij}^{ab}(t)$ according to $\mathbb{P}_t$;
    $\mathbf{t} = t + min\{\mathbf{t_p} | p \in \mathbb{P}_t\}$;
    //rate allocation
    calculate $r_{ij}(t)$ using equation (16)(17);
return $x_{ij}^{ab}(t), r_{ij}(t)$

V. Algorithm design

Since CEOMD problem is NP-hard, we approach the optimal based on some heuristics. The design of our algorithm can be split into three main phases: time slot mergence, path selection, and bandwidth allocation. Then we will claim the design details and the main heuristics of each phase respectively as follows.

1) Time slot mergence: We observe that when there is not much data to transmit, it is totally possible for the transmission process to end before the network topology changes. What's more, even if the topology changes, the transmission process can keep stable as long as there is no path being affected. So one heuristic of our algorithm is that we reduce the original CEOMD problem into multiple simple subproblems in different time durations in which the path set can remain static (For convenience, we call such time durations path-static durations). We assume the data rate and the loss rate remain the same in these path-static durations since no path is affected. Based on this, the path-static duration PSD_{t_0} at time slot t_0 can be formulated as follows:

$$PSD_{t_0} = min\{t | \mathbb{P}_{t_0} \notin \mathcal{E}(t)\} - t_0, \tag{13}$$

where $\mathbb{P}_{t_0}$ is the currently using paths of every source. We perform the path selection and bandwidth allocation at the very beginning of each path-static duration or when some sources finish the transmission process. After that, the path-static duration can be updated.

2) Path selection: The main reason for the long computation time of CEOMD is the infinite number of possible paths for each subflow, let alone the path set could vary at different time slots. So the first thing we do in each path-static duration is to predefine the path set at the very beginning. Since there are many distributed ground stations over the world that can be used to downlink EO data and there are many equal cost shortest paths between any source and ground station pair thanks to the mesh-like property of LEO satellite networks, we will use these paths as our candidate paths to be selected for each subflow. In this way, we can not only expand the number of candidate paths, but also ensure that the paths are not too long.

In addition, intuitively, sources with more data to transmit tend to have longer transmission completion times, so they should have higher priorities while transmitting data in order to reduce the overall completion time. What's more, to accelerate the delivery process, the jointness (the number of paths passing through the same link) between paths should be as low as possible to reduce the bandwidth competition among sources. According to the two heuristics, the newly selected path of a source should have as low jointness with paths of source that have more data to transmit as possible. Besides, when we choose paths, we should also consider the paths' remaining time before being affected by upcoming topology changes (which we also call alive time), the longer the better. Because in this way, it can transmit data stably for as long as possible. To achieve this, we define a metric $\mathcal{M}_p$ as follows to formulate the quality of path p that can provide more potential to have less completion time.

$$\mathcal{M}_p = \frac{\mathbf{t}_p}{\max_{e \in p}\{\sum_{j=1}^{n} R_j^e * d_j\}}, \tag{14}$$

where R_j^e is a 0-1 variable which indicates whether paths of source s_j pass through edge e. Then $\sum_{j=1}^{n} R_j^e * d_j$ represents the jointness with currently selected paths and the sum of priorities of sources that use this link. The smaller this value the better. $\mathbf{t}_p$ is the maximum time duration that path p keeps alive, which can be calculated as follows:

$$\mathbf{t}_p = \min\{t | e \notin \mathcal{E}(t), \forall e \in p\}. \tag{15}$$

The larger $\mathbf{t}_p$ means the source can transmit more data without changing paths. More details of calculating the candidate paths and their alive time are shown in algorithm 1.

Then, we greedily select paths for each flow (sources with large data volumes first) while ensuring that the paths have the highest possible $\mathcal{M}_p$ values. However, in different path-static durations, the routes selected by the greedy strategy may differ significantly. This will lead to large route rescheduling overhead in practice. To avoid this, we only alter the routes which are affected by the topology change but also follow the greedy strategy.

3) Bandwidth allocation: After the selection, the paths are determined for each source. Then we should allocate the bandwidth for each flow/subflow to accelerate the overall data transmission process. As we said before, the sources with more data tend to have a longer completion time. So when paths of different sources pass through the same link, it's feasible to allocate more bandwidth to which has more data left. Specifically, we define b_{ij}^{uv} as the amount of bandwidth allocated to the jth subflow of source s_i on link (u, v), then we proportionally distribute the available bandwidth of link (u, v) to all paths that are using this link,*i.e.*,,

$$b_{ij}^{uv} = \frac{d_i}{\sum_{k=1}^{n} R_j^e * d_k} \mathcal{H}_{uv}, \tag{16}$$

then the final bandwidth for the subflow r_{ij} is as follows:

$$r_{ij} = \min_{(u,v) \in p_{ij}} \{b_{ij}^{uv}\}, \tag{17}$$

where p_{ij} is the path selected for the jth subflow of s_i.

Based on these heuristics, we design a *Heuristic Multipath routing and Bandwidth Allocation (HMRBA)* algorithm. The details of HMRBA are shown in algorithm 2. Line 4-9 indicates that when the path set doesn't change and no source finishes the data transmission, we keep the paths and data rate allocation unchanged. Otherwise, we will first delete the invalid paths that have been affected by topology change, reschedule the affected ones and update the path indicator from line 11-21. After we determine the path of each subflow, we decide the allocation of bandwidth to each subflow by distributing proportionally according to the data volume of each source. Then we send the path and corresponding data rate information to the sources together with the request to control the transmission process. In the next section, we find that our method can achieve much faster data transmission than existing methods.

VI. Performance Evaluation

A. Experiment Setup

Basic component settings. We use two different EO constellations: Dove [2], and SkySat [4] as the data generating sources to declare our method's scalability. They differ in the aspects of altitude, scale, and inclination. All the satellites continuously collect EO data while orbiting the earth according to [24]. As for the broadband constellations, we select the 4th shell StarLink [1], which consists of 5 orbits with 75 satellites each, since its inclination is closest to the EO constellations and it's high enough to provide communication coverage for EO satellites. We inter-connect the starlink satellites following the well-known +Grid topology [6], [16], [20], each of them typically has 4 ISLs connecting the four neighbor satellites in inter-plane and intra-plane, and a GSL accessing the ground stations within the transmission range. We also assume that every EO satellite is equipped with 4 ISL terminals to connect to the nearest 4 starlink satellites (with maximum signal strength). The bandwidth of ISL is set to 2Gbps according to [37], and the maximum downlink data rate of GSL is set to 800Mbps. 173 ground stations provided by [35] are used to configure the distribution of the ground station network. The Goddard Space Flight Center [31] is selected as our destination, which is the EO center owned by NASA. The geolocation of satellites with time is simulated according to their Two-Line-Element (TLE) provided by Celestrak [7] by using python package *ephem* which provides high-precision astronomy computations. Based on the geolocation of each node at each time slot, we build the dynamic network topology according to the visibility between nodes using the python package *networkx*.

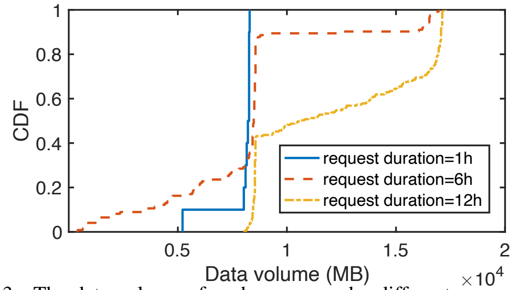

Fig. 3. The data volume of each source under different request time.

AoI and EO data. Here we select the Amazon rainforest (10°N-10°S,73°W-40°W) as our monitoring target, in which a large-scale wire fire happened in 2019. We simulate the cases in which we need the past 1 hour, 6 hours and 12 hours of target area's data collected by EO constellations respectively, which we call the request duration. Here we use the 1-hour request duration to simulate urgent cases like wire fire monitoring and the other two to simulate scenarios which are not that pressing like weather forecasting. What's more, different request durations means different number of sources with requested data and different data volume onboard. In our simulation, there are 10, 128, and 197 sources with data of target AoI when the request duration is 1/6/12 hours respectively. According to [22], the dove satellite takes a picture per second. Based on the image product information provided by [23], we can get the data generating rate of dove satellites. Then, we calculate the amount of EO data on each source based on the time EO satellites fly over the target area and the data generation rate, which is plotted as Fig.3.

Performance comparison. To verify the effectiveness of our algorithm, we compare our method with the following state-of-the-art schemes: (1) Download all EO data of the target area without satellite networks, only via distributed ground stations, like L2D2 [48]; (2) Single path routing for valuable

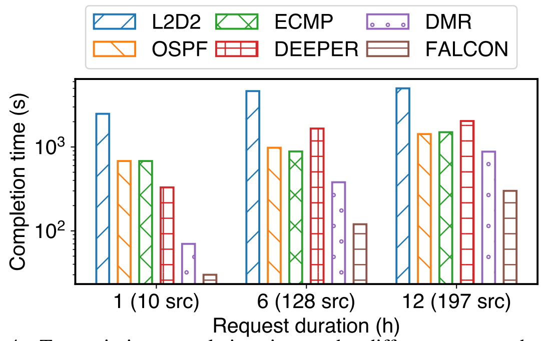

Fig. 4. Transmission completion time under different request duration.

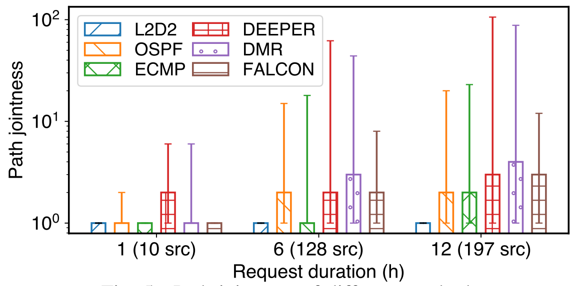

Fig. 5. Path jointness of different methods.

EO data, like in [25], which exploits only one path in the LEO satellite network for every source to transmit data, we use OSPF to emulate the routing process. (3) Single path but with traffic engineering, here ECMP [19], while equally sharing bandwidth if different paths pass through the same link. (4) Recently proposed routing algorithm based on SDN: aDaptive satEllitE-ground cooPerativE tRansmission (DEEPER) [42], which aims to achieve better performance by being aware of network workloads. (5) Disjoint multipath routing (DMR) [28] with fairly shared bandwidth which is a method widely used in Ad Hoc networks. We complete these approaches in Python. And the transmission process is simulated on a Linux server with 40 Intel Xeon ES-2630 v4 CPU cores (2.2GHz) and 32 GB DDR4 RAM.

B. Evaluation of Our Architecture's Performance

Data download completion time. Fig.4 gives the overall delivery completion time when the data request duration is 1/6/12 hours if each source uses 4 paths simultaneously. L2D2 needs hours to finish the EO data downloading since the sources have to wait for visible ground stations. By exploiting satellite networks, the download time can be accelerated by 10-100 folds since we can transmit the data no matter where the source is. When we use the single shortest path (OSPF), although the completion time is nearly ten times smaller than L2D2, we find that there are 10s of paths sharing the same bottleneck link inducing severe bandwidth competition when the number of sources increases as shown in Fig.5. ECMP performs just a little better than OSPF because the sources that have the requested data of target AoI tend to be geographically close, which means that path contention will occur as long as it is based on the shortest path, although it can do some load balancing between multiple shortest paths. Since DEEPER is inclined to choose those paths with more usable capacity, they can achieve better performance when the number of sources

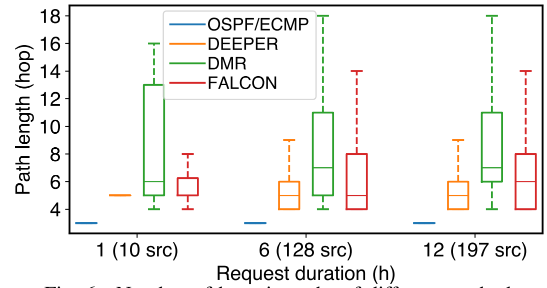

Fig. 6. Number of hops in paths of different methods.

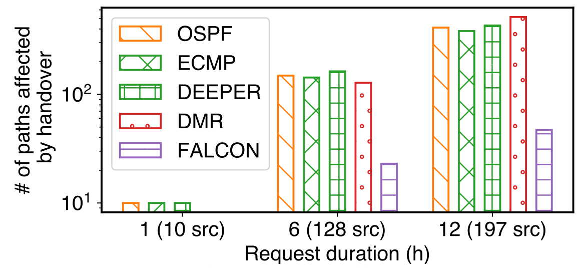

Fig. 7. The number of paths that are affected by handover events.

is small. However, every source doesn't consider the existence of others causing many sources all choosing the path with the maximum capacity which further induces heavy bandwidth competition when the number of sources increases. That is why its performance is even worse than OSPF when the request duration is 6h or 12h. In DMR, by using disjoint multipath, more bandwidth resources are utilized, and there is no path contention among concurrent paths of each source. Hence, the data download time is reduced to even 1 minute when the number of sources is small. However, as the number of sources increases, the inevitable bandwidth competition between different sources becomes more intense since it uses more paths but doesn't consider the path jointness among different sources. From Fig.5, there are some links with more than 50 or even nearly 100 paths passing through them when the number of sources scales up, causing serious bandwidth contention. Our FALCON outperforms the DMR by 51%-58% since we consider the path jointness among sources while selecting paths to improve the overall throughput and we also allocate the bandwidth resources appropriately according to the data volume of each source. As shown in Fig.5, the maximum jointness among paths of FALCON doesn't exceed 10, which is nearly 10 times smaller than that of DMR. In addition, when selecting paths, priority is given to paths with longer survival times so that most paths can complete transmission before handover occurs.

Path length and effect of handovers. To claim the effectiveness of our method, we exploit the path length of each source in each state-of-the-art routing method as shown in Fig.6. It's easy to find that, to ensure the concurrent paths of each source are disjoint, DMR gets the highest path length which is up to 18 hops when there are many sources. In contrast, the average path length of our FALCON is only 3-4 hops more than that of OSPF/ECMP which uses the shortest paths. This is because although we don't choose the shortest paths to the destination, we still choose the shortest path to the ground stations instead. Since DEEPER tends to use paths

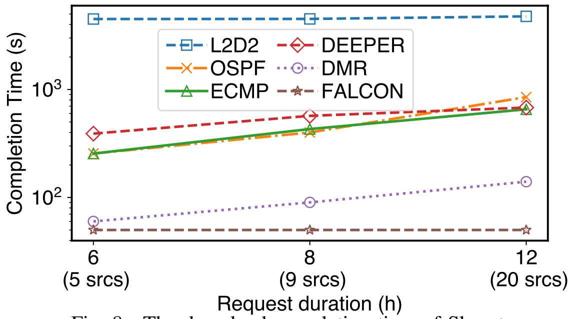

Fig. 8. The download completion time of Skysat.

with more capacity, the path length is slightly more than that of OSPF/ECMP. In addition, we show the impact of handover events on the transmission process by recording the number of path changes during the transmission as shown in Fig.7. When there are only 10 sources, the DMR and FALCON can both finish the transmission process before handover happens, so the number of paths affected by handover during the transmission process is zero while the OSPF/ECMP/DEEPER suffers nearly 10 path changes on average. As the number of sources increases, the path changes of DMR increase dramatically and even surpass that of OSPF/ECMP/DEEPER since it uses more paths simultaneously which are inevitably affected when the delivery time exceeds the maximum visible time between satellites and ground stations. Instead, the number of paths affected by handover in our FALCON remains the lowest, because we tend to choose paths with more alive time considering the mobility of the network topology. In addition, by fine-tuning the routing and bandwidth allocation process, most of the sources can finish the transmission before the handover event happens.

Adaptation of different EO constellations. To demonstrate the scalability of our method, a smaller inclined orbit EO constellation–Skysat, which contains 20 satellites, is also tested in our experiment since the visible time between EO satellites and broadband satellites could vary when the inclination and altitude of satellites are different. As in Fig.8, we find that the data delivery completion time remains the lowest using FALCON, mainly because our FALCON can make fuller use of the bandwidth resources by controlling the path overlap and reducing overall delivery completion time by fine-tuning the bandwidth allocation for such relatively small EO constellation. The delivery completion time of DMR is relatively longer just like in Dove. Another interesting phenomenon is that the download completion time of L2D2 also maintains nearly the same level because the time duration waiting for visible ground stations makes up most of the completion time, which takes nearly more than one hour.

VII. Other Related Work

In addition to prior solutions introduced in §II-C, we discussed other efforts related to our study in this paper.

Multipath routing. Plenty of research focuses on multipath routing and load balancing. In data center networks, ECMP [19] is commonly deployed to achieve good load balancing by distributing traffic equally over multiple paths with the same cost using a simple round-robin pattern. But the overall throughput is limited by the path with minimum capacity. WCMP [52] was proposed to distribute load flexibly to alleviate the problem of ECMP. The paths used by one source are not guaranteed to be disjoint, so the competition of bandwidth within one source may happen especially in such a mesh-like satellite network. In mobile ad-hoc networks, a number of multipath routing protocols are proposed in order to increase the reliability while transmitting data (*e.g.*,, fault tolerance) or load balancing. Two presentative protocols are Split Multipath Routing (SMR) [27], Ad hoc On-demand Multipath Distance Vector (AOMDV) [30]. SMR is designed to find maximally disjoint paths between one source and destination pair by exploiting dynamic source routing. AOMDV is an extension to the AODV [33] protocol for computing loop-free and link-disjoint paths. A recent work [43] designed a similar multipath routing protocol but combined it with network coding to achieve better performance of multipath EO data transmission in satellite networks. However, none of them considers the path relation among different sources. Path contention undermines the advantages of multipath transmission. Our routing scheme improves the multipath transmission performance by considering the path correlation among different sources.

Coflow scheduling. Many coflow scheduling works that aim to minimize the coflow completion time are proposed in data center networks. [51] proposes a coflow-aware network optimization framework that seamlessly integrates routing and scheduling for better application performance. [29] studies the routing and scheduling of multiple coflows to minimize the average coflow completion time. [38] also proposes algorithms to deal with single coflow scheduling and multiple coflow scheduling problems. However, there are two main differences between the coflow scheduling problem in data center networks and our CEOMD problem. Firstly, unlike data center networks, the topology of LEO satellite networks changes with time since the high mobility of satellites. Second, we exploit multipath for each flow. All of these contribute to the difficulty to solve the CEOMD problem.

VIII. Conclusion

In this paper, we present FALCON, which makes full use of multipath in mesh-like LEO satellite networks to accelerate the constellation-wide EO data delivery process, especially for urgent cases. We first formulate the Constellation-wide EO data Multipath Download (CEOMD) problem which is proved to be NP-hard. Then we propose a heuristic multipath routing and bandwidth allocation algorithm to ensure low path contention among sources and shorten the overall data delivery completion time. Extensive evaluations show that our method is scalable and can be at least 51% faster than the state-of-the-art methods under different constellation settings.

IX. Acknowledgment

We thank all reviewers for their feedback which greatly improved the paper. This work was supported by the National Key R&D Program of China (No. 2022YFB3105202), National Natural Science Foundation of China (NSFC No. 62132004), and Tsinghua University-China Telecom Joint Research Institute for Next Generation Internet Technology.

REFERENCES

[1] Spacex, planet ink deal to launch earth-imaging satellites through 2025. https://www.space.com/spacex-planet-satellite-launch-deal-2025.

[2] E. S. Agency. planetscope. https://earth.esa.int/eogateway/missions/planetscope.

[3] E. S. Agency. Rapideye. https://earth.esa.int/eogateway/missions/rapideye.

[4] E. S. Agency. skysat. https://earth.esa.int/eogateway/missions/skysat.

[5] Amazon. Amazon kuiper. https://www.geekwire.com/2019/amazon-project-kuiper-broadband-satellite/.

[6] D. Bhattacherjee and A. Singla. Network topology design at 27,000 km/hour. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pages 341–354, 2019.

[7] CelesTrak. Norad gp element sets current data. http://celestrak.org/NORAD/elements/.

[8] I. del Portillo, B. Cameron, and E. Crawley. Ground segment architectures for large leo constellations with feeder links in ehf-bands. In *2018 IEEE Aerospace Conference*, pages 1–14. IEEE, 2018.

[9] B. Denby and B. Lucia. Orbital edge computing: Nanosatellite constellations as a new class of computer system. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 939–954, 2020.

[10] K. Devaraj, M. Ligon, E. Blossom, J. Breu, B. Klofas, K. Colton, and R. Kingsbury. Planet high speed radio: Crossing gbps from a 3u cubesat. 2019.

[11] ESA. Copernicus: Sentinel-2. https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/copernicus-sentinel-2.

[12] ESA. Gaofen-1. https://earth.esa.int/web/eoportal/satellite-missions/g/gaofen-1.

[13] ESA. High resolution optical imagers. http://www.eohandbook.com/eohb2011/earth_imagers.html.

[14] R. Frost. What's the time delay in communicating from iss to someone on earth? https://www.quora.com/Whats-the-time-delay-in-communicating-from-ISS-to-someone-on-Earth.

[15] G. Furano, G. Meoni, A. Dunne, D. Moloney, V. Ferlet-Cavrois, A. Tavoularis, J. Byrne, L. Buckley, M. Psarakis, K.-O. Voss, et al. Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities. *IEEE Aerospace and Electronic Systems Magazine*, 35(12):44–56, 2020.

[16] M. Handley. Delay is not an option: Low latency routing in space. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks (HotNets)*, pages 85–91, 2018.

[17] M. Handley. Using ground relays for low-latency wide-area routing in megaconstellations. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks (HotNets)*, pages 125–132, 2019.

[18] F. Heine, H. Kämpfner, R. Lange, R. Czichy, M. Lutzer, and R. Meyer. Laser communication applied for edrs, the european data relay system. *CEAS space journal*, 2(1):85–90, 2011.

[19] C. Hopps. Analysis of an equal-cost multi-path algorithm. Technical report, 2000.

[20] M. K. Jonathan F.C. Herman, Stephen MancePaul A. Forquera. Satellite constellation. us20210070477a1/en. *https://patents.google.com/patent/*, 2021.

[21] P. Lab. Planet imagery product specifications. https://assets.planet.com/docs/Pla-net_Combined_Imagery_Product_Specs_letter_screen.pdf.

[22] P. Labs. Overview of the planet labs constellation of earth imaging satellites. https://www.itu.int/en/ITU-R/space/workshops/2015-prague-small-sat/Presentations/Planet-Labs-Safyan.pdf.

[23] P. Labs. Planet imagery product specifications. https://assets.planet.com/docs/combined-imagery-product-spec-april-2019.pdf.

[24] P. Labs. Planet labs specifications: Spacecraft operations & ground systems. http://content.satimagingcorp.com.s3.amazonaws.com/media/pdf/Dove-PDF-Download.

[25] Z. Lai, Q. Wu, H. Li, M. Lv, and J. Wu. Orbitcast: Exploiting mega-constellations for low-latency earth observation. In *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, pages 1–12. IEEE, 2021.

[26] T. Lan, H. Li, Q. Wu, Z. Lai, and J. Liu. Exploiting path diversity to increase system performance in mega-constellations. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–7. IEEE, 2021.

[27] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *2001 IEEE International Conference on Communications (ICC).*, volume 10, pages 3201–3205. IEEE, 2001.

[28] X. Li and L. Cuthbert. On-demand node-disjoint multipath routing in wireless ad hoc networks. In *29th annual IEEE international conference on local computer networks (LCN)*, pages 419–420. IEEE, 2004.

[29] Y. Li, S. H.-C. Jiang, H. Tan, C. Zhang, G. Chen, J. Zhou, and F. C. Lau. Efficient online coflow routing and scheduling. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 161–170, 2016.

[30] M. K. Marina and S. R. Das. On-demand multipath distance vector routing in ad hoc networks. In *2001 IEEE 9th International Conference on Network Protocols (ICNP)*, pages 14–23. IEEE, 2001.

[31] NASA. Goddard space flight center. https://www.nasa.gov/goddard.

[32] I. of Optical Sensor Systems. High resolution earth observation satellites. https://www.dlr.de/os/en/desktopdefault.aspx/tabid-7155/11913_read-28137/.

[33] C. Perkins, E. Belding-Royer, and S. Das. Rfc3561: Ad hoc on-demand distance vector (aodv) routing, 2003.

[34] Planet. Our constellation. https://www.planet.com/our-constellations/.

[35] SatNOGS. Open source global network of satellite ground-stations. https://satnogs.org/.

[36] SentinelHub. Planetscope. https://docs.sentinel-hub.com/api/latest/data/planet-scope.

[37] M. I. Shehata, M. A. Zaki, H. A. Fayed, A. Abd El Aziz, and M. H. Aly. Optical inter-satellite link over low earth orbit: Enhanced performance. In *Journal of Physics: Conference Series*, volume 2128, page 012001. IOP Publishing, 2021.

[38] L. Shi, Y. Liu, J. Zhang, and T. Robertazzi. Coflow scheduling in data centers: routing and bandwidth allocation. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 32(11):2661–2675, 2021.

[39] Skywatch. What resolution do i need when using satellite earth observation data? https://www.skywatch.com/blog/what-resolution-do-i-need-when-using-satellite-earth-observation-data.

[40] T. Sobchak, D. W. Shinners, and H. Shaw. Nasa space network project operations management: Past, present and future for the tracking and data relay satellite constellation. In *2018 SpaceOps Conference*, page 2358, 2018.

[41] SpaceX. Starlink. https://www.starlink.com/.

[42] F. Tang. Dynamically adaptive cooperation transmission among satellite-ground integrated networks. In *In 2020 IEEE Conference on Computer Communications (INFOCOM)*, pages 1559–1568. IEEE, 2020.

[43] F. Tang, H. Zhang, and L. T. Yang. Multipath cooperative routing with efficient acknowledgement for leo satellite networks. *IEEE Transactions on Mobile Computing*, 18(1):179–192, 2018.

[44] E. union agency for the space programme. Earth observation market. https://www.euspa.europa.eu/european-space/euspace-market/earth-observation-market.

[45] UnionofConcernedScientists. Ucs satellite database. https://www.ucsusa.org/resources/satellite-database.

[46] USGS. Landsat-4. https://www.usgs.gov/landsat-missions/landsat-4.

[47] D. Vasisht and R. Chandra. A distributed and hybrid ground station network for low earth orbit satellites. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks (HotNets)*, pages 190–196, 2020.

[48] D. Vasisht, J. Shenoy, and R. Chandra. L2d2: low latency distributed downlink for leo satellites. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 151–164, 2021.

[49] J. G. Walker. Satellite constellations. *Journal of the British Interplanetary Society*, 37:559, 1984.

[50] A. K. Whitcraft, I. Becker-Reshef, B. D. Killough, and C. O. Justice. Meeting earth observation requirements for global agricultural monitoring: An evaluation of the revisit capabilities of current and planned moderate resolution optical earth observing missions. *Remote Sensing*, 7(2):1482–1503, 2015.

[51] Y. Zhao, K. Chen, W. Bai, M. Yu, C. Tian, Y. Geng, Y. Zhang, D. Li, and S. Wang. Rapier: Integrating routing and scheduling for coflow-aware data center networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 424–432. IEEE, 2015.

[52] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat. Wcmp: Weighted cost multipathing for improved fairness in data centers. In *Proceedings of the Ninth European Conference on Computer Systems*, pages 1–14, 2014.