On the Fine-Grained Distributed Routing and Data Scheduling for Interplanetary Data Transfers

Xiaojian Tian and Zuqing Zhu, *Fellow, IEEE*

***Abstract*—Interplanetary networks (IPNs) are complex communication infrastructures used for data exchange among spacecrafts, rovers and ground stations. Due to the significant delay and uncertainty in communications, efficient routing and data scheduling of interplanetary data transfer (IP-DT) becomes crucial. With the increase of deep space (DS) exploration missions, it would be difficult for existing IPNs to cope with the growing of IP-DT demands. In this work, to improve the performance of IP-DTs in IPNs, we formulate an integer linear programming (ILP) model and design an effective fine-grained distributed routing and data scheduling (FD-RDS) algorithm based on it. We prove that the proposed algorithm is a polynomial-time 2-approximation algorithm for solving the ILP model. Extensive simulations show that our proposals can significantly improve the efficiency and reliability of IPNs. Specifically, our proposals outperforms known benchmarks in terms of both the delivery ratio and E2E latency of IP-DTs.**

***Index Terms*—Interplanetary network, Delay tolerant network, Distributed route selection, Distributed scheduling.**

I. Introduction

OVER past decades, the fast development of the Internet has penetrated almost everywhere on and around our planet [1–3]. Recently, deep space (DS) exploration and interplanetary science missions have gained increasing interests from government, industry and academia [4, 5], which has created the need of expanding the Internet to include interplanetary networks (IPNs) [6]. As shown in Fig. 1, IPN is the infrastructure to enable the communications among DS objects, including the ground stations, satellites, landers, rovers, and so on [7], and thus it is crucial for DS missions.

IPN has a few unique characteristics, making it significantly different from the networks on Earth. First, each IPN has a dynamic and unstable topology, because the movement and shields of DS objects and celestial bodies will cause the connections among them (*i.e.*, links in the IPN) to be intermittent. This complicates the routing and resource allocation in IPNs and makes existing techniques [8–12] inapplicable, as end-to-end routing paths are unavailable for communication sessions. Second, as the motion of celestial bodies is usually predictable and the orbits of DS objects are planned, we should consider these predictable topology changes when routing traffic and scheduling data transfers in an IPN, resulting in time-sensitive network control and management (NC&M). Finally, because the average link length in IPNs can be several orders of magnitude longer than that on Earth, the resulting extremely long communication delay will make it difficult for nodes to exchange state information, or even if the information can be got, it will not be up-to-date. Hence, the research and development (R&D) of IPN technologies are facing some unexplored challenges. For instance, it would be infeasible to architect IPNs with software-defined networking (SDN) [13–15], which suggests that distributed NC&M will be inevitable.

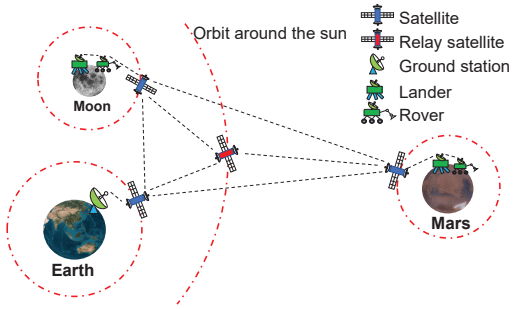


Fig. 1. An example of IPN.

The R&D on IPNs is still in its early stage, and thus current IPNs usually use relatively simple architectures with sparse topology density. Moreover, current IPNs were usually built specifically for certain DS missions, and thus they do not interoperate to work as the networks in the Internet. However, the rapid development and future plans of DS missions in recent years will bring more DS objects and network services into IPNs, which will not only expand their topology scales but also increase the traffic loads in them. Therefore, for the sake of cost-effectiveness, future IPNs will be integrated with a unified Internet infrastructure [16], instead of working as physically-unconnected autonomous systems.

Existing IPN routing and data scheduling schemes are difficult to meet the challenges brought by future IPN network environment. On one hand, existing routing schemes [17] ignore the queuing delay on IPN nodes and just let the bundles, which are the atomic data units for interplanetary data transfers (IP-DTs), be forwarded sequentially, lacking effective bundle scheduling. This may not be an issue in the sparse environment of current IPN networks, but as future IPNs will have much denser topologies and heavier traffic loads, the queuing delay on nodes cannot be ignored anymore and would greatly reduce the performance of IP-DTs otherwise [18]. On the other hand, even though certain existing schemes [19–22] did consider routing together with data scheduling for IP-DTs, they only calculate the routing path for each bundle once based on the current or estimated network status, lacking fine-grained adjustments on-the-fly. These drawbacks of existing

X. Tian and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (email: zqzhu@ieee.org).

Manuscript received on March 31, 2023.

mechanisms motivated us to design more efficient distributed routing and data scheduling methods for IP-DTs.

In this paper, we propose effective distributed routing and data scheduling algorithms that not only optimize the routing path of each bundle on-the-fly but also realize bundle-level scheduling on each IPN node to maximize the delivery ratio of IP-DTs as well as minimizing their average end-to-end (E2E) latency. We first formulate an integer linear programming (ILP) model to solve the routing and data scheduling on each IPN node in a future period of time exactly to obtain the optimal IP-DT scheme for the node. Then, we design a 2-approximation polynomial-time algorithm to solve the problem modeled by the ILP time-efficiently, for coping with large-scale scenarios. Extensive simulations verify the performance of our proposals. Specifically, our distributed routing and data scheduling algorithms outperform known benchmarks in terms of both the delivery ratio and E2E latency of IP-DTs.

The rest of the paper is organized as follows. Section II introduces the background of IPN and gives a brief survey on the related work. We explain the network model, and formulate and solve the optimization of fine-grained distributed routing and data scheduling in an IPN in Section III. Then, the design of our approximation algorithm is presented in Section IV. We evaluate our proposals with numerical simulations in Sections V. Finally, Section VI summarizes the paper.

II. Background and Related work

A. Background

The challenges faced by IPNs, *e.g.*, extremely-long transmission delay and unstable connections, can be addressed by leveraging delay tolerant networking (DTN) [23], which adopts "store-carry-forward (SCF)" to realize IP-DTs. Specifically, DTN groups a set of continuous data blocks as a *bundle*, and uses it as the atomic unit for IP-DTs. Therefore, each bundle contains sufficient information for being routed to its destination with SCF and driving its application running there to make progress [24]. The bundle-based DTN protocols have been standardized in [24, 25]. However, as IPN has unique characteristics and is not just an ordinary DTN (*i.e.*, we need to consider the future contact plan of each link to route and schedule IP-DTs [16]), agents such as the National Aeronautics and Space Administration (NASA) have developed specific techniques for IP-DTs and implemented related DS tests. One famous example is the contact graph routing (CGR) algorithm [17], which routes IP-DTs by calculating the routing path for each bundle once based on the time-varying topology of an IPN and the information about the bundle (*e.g.*, its size, priority and expiration time). Meanwhile, the Jet Propulsion Laboratory (JPL) has developed the interplanetary overlay network (ION) [26, 27], which is an open-source software platform for emulating the bundle-based protocols for IPNs.

B. Related Work

Routing and data scheduling has always been important for IPNs, due to the poor network connectivity in them. NASA has proposed CGR [17] for routing IP-DTs based on the contact plan of an IPN, and has provided a detailed tutorial to describe the implementation details [28]. Nevertheless, CGR ignores the queuing delay of bundles and assumes that pending bundles can be sent immediately upon each contact of a link. The assumption is only valid in an IPN whose traffic load is very low, but in the future when an IPN is heavy-loaded, a number of bundles can be buffered at each node and thus the queuing delay caused by them will become long enough to make certain bundles miss their transmission opportunities during contacts. To address this issue, people have developed a few enhanced versions of CGR [19–22], among which the CGR-ETO in [21] is a representative one, because it improves the prediction accuracy of bundle delivery time by taking the estimated queuing delay of bundles into account. A modified temporal graph model was introduced in [22], based on which the earliest arrival optimal delivery ratio (EAODR) routing algorithm was proposed. In [29], the authors considered concurrent transmission of bundles, and proposed a bundle management scheme that assigns data-rates to bundles based on link capacities and buffer resources on IPN nodes to avoid buffer overflow. However, all of these schemes process bundles in queues in the first-in-first-out (FIFO) way, but do not consider bundle-level scheduling in each queue.

Note that, the extremely long communication delay in IPNs makes distributed NC&M inevitable, but due to their unique characteristics, the distributed routing and data scheduling in IPNs cannot be directly solved with the schemes developed for the networks on/around Earth [30–32]. Hence, in [33], the authors proposed the Multi-Attribute Routing and Scheduling (MARS) algorithm, which addresses bundle-level scheduling in queues and thus can achieve better performance on IP-DTs than CGR and its enhanced versions. Nevertheless, MARS only optimizes the transfers of bundles in one queue greedily, but does not try to schedule bundles in multiple queues on a node jointly. Note that, multiple queues can be created on each IPN node, to correspond to different next hops, network applications, and service priorities, especially in the future IPNs that have denser connectivity, heavier traffic loads, and more diversified applications. Therefore, we considered how to schedule bundles in multiple queues on a node jointly, designed an online routing and data scheduling algorithm in [34] by leveraging the Lyapunov optimization [35], and verified that better performance on IP-DTs can be achieved over MARS. However, the approach in [34] still treats routing and data scheduling separately. Hence, the algorithm designed in [34] can still be improved if a more rigorous model will be formulated to optimize routing and data scheduling jointly.

As we will explain later, it is $\mathcal{NP}$-hard to optimize routing and data scheduling in an IPN jointly, and thus finding its exact solution can be computationally intractable [36], especially when the problem size is large. Therefore, inspired by the studies in [37–39], we resort to designing an approximation algorithm to find a near-optimal solution for it time-efficiently.

III. Problem Formulation

In this work, we consider fine-grained distributed routing and data scheduling (FD-RDS) on each node in an IPN to maximize the delivery ratio of IP-DTs as well as minimizing

their average E2E latency. As we will explain in the following, our FD-RDS can optimize the tradeoff between the delivery ratio of IP-DTs and their average E2E latency better than the existing approaches in [17, 33, 34] for two reasons: 1) it conducts the optimization in the bundle-level and 2) it optimizes routing and data scheduling jointly. The notations used in this section are summarized in Table I.

TABLE I
NOTATIONS USED IN PROBLEM FORMULATION

Notation	Description
G^t	the time-vary graph to represent IPN's topology
V	the set of IPN nodes
E^t	the set of temporal links at time t
E^t_u	the set of links that originate from u
e^t	a temporal link
u, v	the end nodes of a temporal link
t^s, t^e	the start and end time of the contact of a temporal link
r	the data-rate of a temporal link
τ	the transmission latency of a temporal link
B	a bundle for IP-DT
s, d	the source and destination of a bundle
β	the data size of a bundle
t^a, t^d	the generated time and deadline of a bundle
q	the priority of a bundle
$\mathcal{Q}_v$	the queue on node v
$\mathcal{Q}_{v,u_1}$	the outgoing queue on node v for bundles to node u_1
TS	the time slot
T_0	the scheduling period
t_0	the start time of a scheduling period
$x_{B,e^t,t}$	the boolean variable for IP-DT of bundle B on link e^t
f_1, f_2	the cost functions
t^d_{B,e^t}	the projected delivery time of bundle B through e^t
t^e_{B,e^t}	the expire time of sending bundle B out through e^t
$\hat{\tau}^d_{B,e^t}$	the upper-limit of transmission time of bundle B
τ_{B,e^t}	the transmission time taken by bundle B over e^t
$m_0, \cdots, m_4$	the coefficients for normalization
M, Π	large positive constants
y, y'	binary auxiliary variables

A. Network Model

We model the topology of an IPN as a time-vary graph $G^t(V, E^t)$, where V is the set of IPN nodes, and E^t denotes the set of temporal links at time t. In the following, we use E^t_u to denote the set of links that originate from u. Each temporal link in E^t is defined as $e^t(u, v, t^s, t^e, r, \tau)$, where u and v are its end nodes ($u, v \in V$), t^s and t^e denote the start and end time of its contact, respectively, r is its data-rate, and τ is its transmission latency. A bundle for IP-DT is represented as $B(s, d, \beta, t^a, t^d, q)$, where s and d are its source and destination, respectively, β denotes its data size, t^a is the time when it is generated at the source s, t^d is the deadline by when it should be sent to the destination d, and q denotes the priority of its IP-DT according to the bundle's application.

B. Fine-Grained Distributed Routing and Data Scheduling

We first introduce the overall procedure of distributed routing and data scheduling in an IPN to facilitate the subsequent optimization formulation and algorithm design. As shown in Fig. 2, we allocate a queue $\mathcal{Q}_v$ on each node $v \in V$ in the IPN to buffer outgoing bundles, which include the bundles

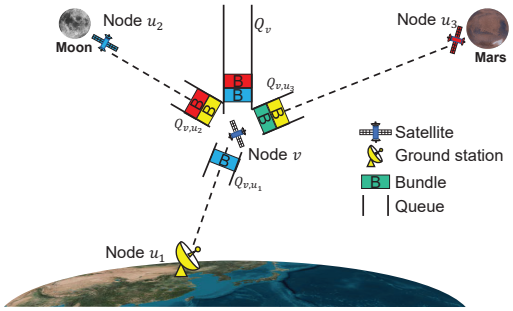


Fig. 2. Fine-grained distributed routing and data scheduling in an IPN.

using v as an intermediate node and those being generated locally. Meanwhile, for each link that originates from v, we also allocate a queue to store the bundles that are scheduled from $\mathcal{Q}_v$ and will be transmitted in sequence during future contacts. For example, the queue $\mathcal{Q}_{v,u_1}$ stores the bundles that will be transmitted on *Link* $v \rightarrow u_1$ during future contacts. Hence, on each node $v \in V$, FD-RDS moves bundles from $\mathcal{Q}_v$ to outgoing FIFOs (*e.g.*, $\mathcal{Q}_{v,u_1}$) adaptively to maximize the delivery ratio of IP-DTs as well as minimizing their average E2E latency. Due to the extremely long communication latency among IPN nodes, the routing and data scheduling of IP-DTs has to be distributed, *i.e.*, the bundles in each queue $\mathcal{Q}_v$ should only be scheduled based on the local information of node v.

At each contact of a time-varying link $e^t \in E^t$ that is from node v, the resource for scheduling IP-DTs is its transmission capacity in terms of data volume (*i.e.*, $r \cdot (t^e - t^s)$, where r, t^s and t^e are the link's data-rate and the start and end time of the contact, respectively). And in order to limit the scale of the optimization model for FD-RDS, we formulate it to only tackle the scheduling of IP-DTs in T_0 future time slots (TS')[1] each time. Hence, if the start time of a scheduling period is t_0, the period will cover $t \in [t_0, t_0 + T_0 - 1]$ TS'.

Algorithm 1 shows the proposed procedure for FD-RDS. At $t = 0$, we initialize $\mathcal{Q}_v$ and the outgoing FIFOs $\{\mathcal{Q}_{v,u}\}$ at node v (*Line* 1). Then, if t is the start time of a new scheduling period, we obtain the IP-DT schemes of the bundles in $\mathcal{Q}_v$ by inputting the bundles and the contact plan of all the links that will be in contact from node v in the scheduling period to the optimization model and solving it (*Line* 5). The optimization model and how to solve it will be explained later. The solution of the optimization tells the IP-DT scheme of each bundle $B \in \mathcal{Q}_v$, which specifies which $\mathcal{Q}_{v,u}$ the bundle should be enqueued or it should stay in $\mathcal{Q}_v$ (*i.e.*, routing path selection) and its transmission time (or transmission order) in $\mathcal{Q}_{v,u}$ (*i.e.*, data scheduling). Next, *Lines* 6-15 move bundles from $\mathcal{Q}_v$ to $\{\mathcal{Q}_{v,u}\}$, according to their IP-DT schemes. The bundles in each $\mathcal{Q}_{v,u}$ will then be transmitted according to their scheduled time and priorities, and the expired bundles in $\mathcal{Q}_v$ will be removed at the end of each TS (*Lines* 17-22). Note that, due to the feasibility guarantee of the optimization model and our algorithms to solve it, all the bundles in $\{\mathcal{Q}_{v,u}\}$ should be

[1]In this work, we assume that each IPN node works as a discrete-time system operating on TS', each of which has a fixed duration of Δt. Hence, the IP-DT of each bundle is scheduled according to TS' ($t = \Delta t, 2\Delta t, \cdots$), which can be normalized as $t \in \{1, 2, \cdots\}$ for simplicity [34].

Algorithm 1: Procedure of FD-RDS in an IPN

$t = 0$, initialize $\mathcal{Q}_v$ and $\{\mathcal{Q}_{v,u}\}$;
while *node* $v \in V$ *is operational* **do**
  insert all newly generated/received bundles in $\mathcal{Q}_v$;
  if t *is the start time of a new scheduling period* **then**
    solve the optimization model to get IP-DT schemes of bundles in $\mathcal{Q}_v$ in T_0 future TS';
    for *each bundle* $B \in \mathcal{Q}_v$ **do**
      if B *is scheduled for transmission* **then**
        move B to the corresponding $\mathcal{Q}_{v,u}$ according to its IP-DT scheme;
      else
        keep B in $\mathcal{Q}_v$;
      end
    end
    for *each queue* $\mathcal{Q}_{v,u}$ **do**
      sort bundles in $\mathcal{Q}_{v,u}$ according to their priorities for IP-DT;
    end
  end
  for *each queue* $\mathcal{Q}_{v,u}$ **do**
    if $e_t = (v, u)$ *is in contact* **then**
      transmit bundles in $\mathcal{Q}_{v,u}$ in sequence;
    end
  end
  $t = t + 1$, remove expired bundles in $\mathcal{Q}_v$;
end

able to be transmitted within the current scheduling period.

Note that, although this paper only considers distributed routing and data scheduling of IP-DTs, exchanging the information on routing and data scheduling among IPN nodes might further improve the performance of IP-DTs. However, in order to extend the distributed approaches developed in this work to address the situation where the aforementioned information exchanges are possible, we still need to solve two problems: 1) an effective scheme needs to be designed for each IPN node to evaluate and utilize the information received from remote nodes, and 2) an efficient information exchange protocol needs to be proposed to minimize unnecessary bandwidth overheads. These two problems will be studied in our future work.

C. ILP Model for FD-RDS

In the following, we formulate an ILP model to tackle FD-RDS on each node of an IPN, whose procedure for obtaining the IP-DT scheme of a scheduling period is shown in Fig. 3.

Variables:

- $x_{B,e^t,t}$: the boolean variable that equals 1 if bundle B finishes its IP-DT on link $e^t \in E_v^t$ at TS $t \in \mathcal{T}$ ($\mathcal{T} = [t_0, t_0 + T_0 - 1]$ is a scheduling period), and 0 otherwise.

If we have $x_{B,e^t,t} = 1$ at TS $t \in \mathcal{T}$, a cost function can be defined to evaluate its impact on the E2E latency of B:

$$f_1(B, e^t, t) = m_0 \cdot (t - t_0) + m_1 \cdot e^t(\tau) + m_2 \cdot \left(t^d_{B,e^t} - t_0\right), \quad (1)$$

where $e^t(\tau)$ denotes the transmission latency of e^t, t^d_{B,e^t} is the projected delivery time of B (*i.e.*, the projected time of

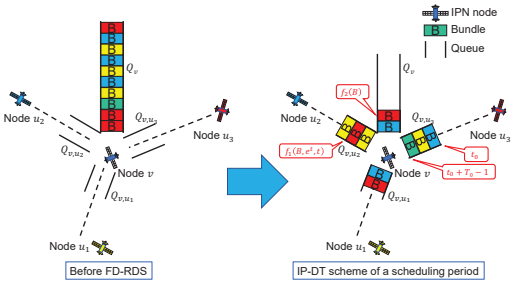


Fig. 3. Obtaining the IP-DT scheme of a scheduling period with FD-RDS.

B reaching its destination by leaving v through e^t, and it can be estimated by leveraging CGR [17]), and m_0, m_1 and m_2 are the coefficients for normalization, respectively. Meanwhile, for the case that B is not scheduled for transmission during $[t_0, t_0 + T_0 - 1]$, we define a penalty function as

$$f_2(B) = m_3 \cdot \min\left[T_0 - 1, B(t^d) - t_0\right] + m_4 \cdot T^d_B, \quad (2)$$

where $B(t^d)$ denotes the deadline of B, T^d_B is the penalty due to the average projected delivery time of B (*i.e.*, if a bundle with earlier projected delivery time is not scheduled, a greater penalty should be imposed), and m_3 and m_4 are also the coefficients for normalization. T^d_B can be obtained as

$$T^d_B = M - \frac{1}{|E^t_v|} \sum_{e^t \in E^t_v} \left(t^d_{B,e^t} - t_0\right), \quad (3)$$

where M is a large positive constant to ensure $T^d_B > 0$.

Optimization Objective:

The optimization objective of FD-RDS is

$$\begin{aligned} \text{Minimize} \quad & \sum_{B \in \mathcal{Q}_v} \sum_{e^t \in E^t_v} \sum_{t=t_0}^{t_0+T_0-1} f_1(B, e^t, t) \cdot x_{B,e^t,t} \\ & + \sum_{B \in \mathcal{Q}_v} f_2(B) \cdot \left(1 - \sum_{e^t \in E^t_v} \sum_{t=t_0}^{t_0+T_0-1} x_{B,e^t,t}\right), \end{aligned} \quad (4)$$

to minimize the cost brought by E2E latencies of scheduled bundles and the penalty caused by unscheduled bundles jointly (at node v and during the current scheduling period). The objective in Eq. (4) can be transformed into

$$\text{Minimize} \quad \sum_{B,e^t,t} \left[f_1(B, e^t, t) - f_2(B)\right] \cdot x_{B,e^t,t} + \sum_B f_2(B), \quad (5)$$

where the second term is a constant that does not depend on the decision variables $\{x_{B,e^t,t}\}$. Therefore, we can remove the second term and further simplify the objective as

$$\text{Minimize} \quad \sum_B \sum_{e^t} \sum_t \left[f_1(B, e^t, t) - f_2(B)\right] \cdot x_{B,e^t,t}. \quad (6)$$

Constraints:

$$\sum_{e^t} \sum_t x_{B,e^t,t} \le 1, \ \forall B \in \mathcal{Q}_v. \quad (7)$$

Eq. (7) ensures that each bundle B at node v can only be scheduled once at most during the current scheduling period.

$$\sum_t t \cdot x_{B,e^t,t} \le \hat{\tau}_{B,e^t}, \ \forall B \in \mathcal{Q}_v, \ \forall e^t \in E^t_v, \quad (8)$$

where $\hat{\tau}^d_{B,e^t}$ denotes the upper-limit of the transmission time taken by a bundle B over link e^t and can be obtained as

$$\hat{\tau}_{B,e^t} = \min\left[B(t^d), e^t(t^e), t^e_{B,e^t}\right], \tag{9}$$

where $B(t^d)$ denotes the deadline of bundle B, $e^t(t^e)$ denotes the end time of the contact of link e^t, and t^e_{B,e^t} denotes the expire time of sending B out through e^t, which is the time by when B will miss its deadline if it still has not finished its IP-DT over e^t and can also be estimated with CGR [17]. Hence, Eq. (8) ensures that if B is scheduled for transmission over e^t during the current scheduling period, the IP-DT can be completed before the upper-limit of its transmission time.

$$\sum_{t'=t_0}^{t}\sum_{B}\tau_{B,e^t}\cdot x_{B,e^t,t'} \le t-t_0+1,\ \forall t\in[t_0,t_0+T_0-1],\ e^t, \tag{10}$$

where τ_{B,e^t} is the transmission time taken by B over e^t:

$$\tau_{B,e^t} = \frac{B(\beta)}{e^t(r)\cdot\Delta t}, \tag{11}$$

where Δt is the duration of a TS. Eq. (10) ensures that the IP-DT periods of bundles using a same link e^t do not overlap, *i.e.*, each link e^t can only transmit a bundle at any given time.

$$\begin{aligned}&\left(q_{B,e^t}-q_{B',e^t}\right)\cdot\left(w_{B,e^t}-w_{B',e^t}\right)\ge 0\\ &\forall e^t,\ t,\ \{B,B'\in\mathcal{Q}_v, B_1\ne B_2\},\end{aligned} \tag{12}$$

where q_{B,e^t} and w_{B,e^t} are defined as

$$\begin{cases} q_{B,e^t} = \sum\limits_{t} B(q)\cdot x_{B,e^t,t},\\ w_{B,e^t} = H+\sum\limits_{t}(t-H)\cdot x_{B,e^t,t}.\end{cases} \tag{13}$$

Here, $B(q)$ is the priority of bundle B and H is a constant that satisfies $H \ge t_0 + T_0$. Note that, if B is not transmitted over e^t, we set q_{B,e^t} as the lowest priority and $w_{B,e^t} = H$ to ensure that Eq. (12) is valid. Therefore, Eq. (12) ensures that for each link $e^t \in E^t_v$, higher-priority bundles will be transmitted earlier than lower-priority ones. As Eq. (12) is nonlinear, we linearize it as follows

$$\begin{cases} q_{B',e^t}-q_{B,e^t}\le(1-y)\cdot\Pi,\\ -y\cdot\Pi\le q_{B',e^t}-q_{B,e^t},\\ w_{B,e^t}-w_{B',e^t}\le(1-y')\cdot\Pi,\\ y\le y',\end{cases} \tag{14}$$

where y and y' are binary auxiliary variables, and Π is a large positive constant.

Theorem 1. *The FD-RDS on each node of an IPN described by the aforementioned ILP model is $\mathcal{NP}$-hard.*

Proof: We prove that the FD-RDS problem is $\mathcal{NP}$-hard with the restriction method in [40]. Specifically, we first reduce it to a special case by applying the following restrictions:

- We set $|E^t_v| = 1$, *i.e.*, node v only has one outgoing link.
- We set $\hat{\tau}^d_{B,e^t} \ge t_0 + T_0, \forall B \in \mathcal{Q}_v,\ \forall e^t \in E^t_v$, which means that during the current scheduling period, the IP-DTs of all the bundles on node v can be completed before the upper-limits of their transmission time.
- We set $m_0 = 0$ in Eq. (1) to make the cost function independent of the bundle's transmission time.
- We make all the bundles have the same priority to relax the constraints in Eq. (12).

Then, the FD-RDS problem can be transformed into one that needs to transmit bundles buffered on node v within a given scheduling period to minimize the objective in Eq. (6). This is equivalent to the general case of the 0-1 knapsack problem, where the length of the scheduling period T_0 represents the knapsack's weight capacity, each bundle B becomes an object with its transmission time τ_{B,e^t} as the weight, and the value of bundle B is defined as $f_2(B) - f_1(B, e^t, t)$. The 0-1 knapsack problem is known to be $\mathcal{NP}$-hard [40], suggesting that the FD-RDS problem is also $\mathcal{NP}$-hard. ■

IV. Approximation Algorithm Design

As the FD-RDS problem described in Section III-C is $\mathcal{NP}$-hard, it will become intractable for large-scale scenarios. Moreover, because certain IPN nodes (*e.g.*, satellites and rovers) might only has very limited computing resources, it would be difficult for them to solve the ILP model quickly even for small/medium-scale FD-RDS problems. Therefore, in this section, we leverage the algorithmic idea in [38] to design a polynomial-time 2-approximation algorithm by relaxing the priority constraints in Eqs. (12)-(14). Specifically, we design the 2-approximation algorithm for FD-RDS based on the local-ratio theorem [41], whose generalized form is as follows.

Theorem 2. *Let $\vec{f}$ be a profit (or penalty) vector and $\mathcal{C}$ be a set of feasibility constraints on vectors $\vec{x}$. Hence, a vector $\vec{x}$ will be a feasible solution to a given problem instance $(\mathcal{C}, \vec{f})$, if it satisfies all the constraints in $\mathcal{C}$. Then, we can decompose $\vec{f}$ into $\vec{f_a}$ and $\vec{f_b}$ as $\vec{f} = \vec{f_a} + \vec{f_b}$. If $\vec{x}$ is a r-approximate solution with respect to both $(\mathcal{C}, \vec{f_a})$ and $(\mathcal{C}, \vec{f_b})$, it is a r-approximate solution with respect to the original problem $(\mathcal{C}, \vec{f})$ [41].*

A. Recursive 2-Approximation Algorithm

We can transform the ILP in Section III-C into a maximization problem by modifying the objective in Eq. (6) as

$$\text{Maximize}\ \sum_{B}\sum_{e^t}\sum_{t}\left[f_2(B) - f_1(B, e^t, t)\right]\cdot x_{B,e^t,t}. \tag{15}$$

Then, we define $f(B, e^t, t) = f_2(B) - f_1(B, e^t, t)$ as the profit of completing the IP-DT of bundle B on e^t at TS t. For the sake of the convenience of algorithm design, we first relax the constraints related to the priorities of bundles (*i.e.*, Eqs. (12)-(14)). In other words, we first design the approximation algorithm for FD-RDS to treat all the bundles buffered on each node v in an IPN equally regardless of their priorities and move them from $\mathcal{Q}_v$ to the outgoing FIFOs $\{\mathcal{Q}_{v,u}\}$, and then sort the bundles in the outgoing FIFOs according to their priorities to ensure that bundles with higher priorities will always be sent out earlier (*Lines* 13-15 in *Algorithm* 1). As we will show in the simulations in Section V, the aforementioned procedure enforces the priorities well and can still make sure that the bundles are scheduled in a near-optimal way. We use $\mathcal{C}$ to denote the set of constraints in the relaxed ILP (*i.e.*, Eqs. (7), (8) and (10)), represent the set of decision variables as $\mathcal{X}$ (*i.e.*, $\mathcal{X} = \{x_{B,e^t,t},\ \forall B \in \mathcal{Q}_v,\ e^t \in E^t_v,\ t \in \mathcal{T}\}$), and define the solution of FD-RDS as S (*i.e.*, $S = \{x_{B,e^t,t} | x_{B,e^t,t} = 1\}$).

Algorithm 2 shows the overall procedure of our approximation algorithm to solve the relaxed ILP, which takes $\mathcal{X}$, $f(B,e^t,t)$, $\mathcal{C}$ as inputs and outputs S. We first calculate $f(B,e^t,t)$ for each variable $x_{B,e^t,t}$ and delete the variable from set $\mathcal{X}$ if we find $f(B,e^t,t)\leq 0$ (*Lines* 1-3). Then, *Lines* 4-6 return $S=\emptyset$ if $\mathcal{X}$ is empty (*i.e.*, all the variables in the original $\mathcal{X}$ have been processed recursively). Next, we randomly select a variable $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ from $\mathcal{X}$ and decompose the profit function $f(B,e^t,t)$ into two parts, $f_a(B,e^t,t)$ and $f_b(B,e^t,t)$ (*Lines* 7-8). Specifically, the decomposition works by defining $f_a(B,e^t,t)$ as follows.

$$f_a(B,e^t,t)=\begin{cases} f(\tilde{B},\tilde{e^t},\tilde{t}), & x_{B,e^t,t}\in\mathcal{A}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right),\\ f(\tilde{B},\tilde{e^t},\tilde{t}), & x_{B,e^t,t}\in\mathcal{I}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right),\\ 0, & \text{otherwise,}\end{cases} \tag{16}$$

where $\mathcal{A}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)=\{x_{B,e^t,t}|B=\tilde{B}\}$, and $\mathcal{I}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)$ is the set of variables $\{x_{B,e^t,t}\}$ that satisfy $B\neq\tilde{B}$ and generate transmission conflict with $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ at $\tilde{t}$ if equaling 1. Specifically, if the start time of transmitting B is $t'=t-(\tau_{B,e^t}-1)$, we have $\mathcal{I}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)=\{x_{B,e^t,t}|B\neq\tilde{B},e^t=\tilde{e^t},t'\leq\tilde{t}\leq t\}$.

Line 9 calls *Algorithm* 2 recursively with the current $\mathcal{X}$, $f_b(B,e^t,t)$ and $\mathcal{C}$ as inputs, and outputs S'. Here, we define a feasible solution of FD-RDS S is "**semi-maximal**" if 1) it contains $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ or 2) it does not contain $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ but adding $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ to it will make the solution infeasible. Hence, for the selected $x_{\tilde{B},\tilde{e^t},\tilde{t}}$, *Lines* 10-14 return a semi-maximal solution.

Algorithm 2: Recursive Approximation Algorithm

Input: $\mathcal{X}$, $f(B,e^t,t)$, $\mathcal{C}$
Output: S

1 **for** *each bundle* $B\in\mathcal{Q}_v$ **do**
2 | remove $x_{B,e^t,t}$ from $\mathcal{X}$ if $f(B,e^t,t)\leq 0$;
3 **end**
4 **if** $\mathcal{X}=\emptyset$ **then**
5 | **return** $S=\emptyset$;
6 **end**
7 select a variable $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ randomly from $\mathcal{X}$;
8 decompose $f(B,e^t,t)=f_a(B,e^t,t)+f_b(B,e^t,t)$;
9 apply *Algorithm* 2 with current $\mathcal{X}$, $f_b(B,e^t,t)$ and $\mathcal{C}$ as inputs and get the output as S';
10 **if** $S'\cup\{x_{\tilde{B},\tilde{e^t},\tilde{t}}\}$ *is a feasible IP-DT scheme* **then**
11 | **return** $S=S'\cup\{x_{\tilde{B},\tilde{e^t},\tilde{t}}\}$;
12 **else**
13 | **return** $S=S'$;
14 **end**

Theorem 3. *For an arbitrary selection of* $x_{\tilde{B},\tilde{e^t},\tilde{t}}$*, the corresponding semi-maximal solution satisfies* 2*-approximation with respect to maximizing* $\sum f_a(B,e^t,t)\cdot x_{B,e^t,t}$.

Proof: For an arbitrary selection of $x_{\tilde{B},\tilde{e^t},\tilde{t}}$, we derive both an upper bound b_{opt} on the optimum objective and a lower bound $b_{\min}$ on the objective achieved by the corresponding semi-maximal solution obtained by *Algorithm* 2. Then, $r=\frac{b_{\text{opt}}}{b_{\min}}$ is an upper bound on the approximation ratio of *Algorithm* 2, which solves a maximization problem.

First, we consider the optimal solution of the relaxed ILP. According to the definition of $f_a(B,e^t,t)$ in Eq. (16), only the variables $\left\{x_{B,e^t,t}\in\mathcal{A}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)\cup\mathcal{I}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)\right\}$ can contribute to the objective function $\sum\limits_{B,e^t,t} f_a(B,e^t,t)\cdot x_{B,e^t,t}$. Then, according to the constraint in Eq. (7), we have

$$\sum_{\left\{B,e^t,t|x_{B,e^t,t}\in\mathcal{A}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)\right\}} f_a(B,e^t,t)\cdot x_{B,e^t,t}\leq f(\tilde{B},\tilde{e^t},\tilde{t}). \tag{17}$$

Specifically, a bundle $\tilde{B}$ only needs to be transmitted once. Similarly, based on Eq. (10), we have

$$\sum_{\left\{B,e^t,t|x_{B,e^t,t}\in\mathcal{I}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)\right\}} f_a(B,e^t,t)\cdot x_{B,e^t,t}\leq f(\tilde{B},\tilde{e^t},\tilde{t}), \tag{18}$$

i.e., the IP-DT periods of bundles using a same link cannot overlap. Eqs. (17) and (18) suggest $b_{\text{opt}}=2\cdot f(\tilde{B},\tilde{e^t},\tilde{t})$.

Then, we move to the semi-maximal solution obtained by *Algorithm* 2. According to the definition of such a semi-maximal solution, it at least contains a variable $\{x_{B,e^t,t}=1\}$ that is in either set $\mathcal{A}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)$ or set $\mathcal{I}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)$. Hence, we obtain $b_{\min}=f(\tilde{B},\tilde{e^t},\tilde{t})$, and prove that the approximation ratio of *Algorithm* 2 is at most $r=\frac{b_{\text{opt}}}{b_{\min}}=2$. ■

Theorem 4. Algorithm *2 is a* 2*-approximation algorithm for the relaxed ILP.*

Proof: First of all, *Lines* 1-3 in *Algorithm* 2 do not affect its optimality, because the relaxed ILP is for maximization and they only remove $x_{B,e^t,t}$ from $\mathcal{X}$ if it leads to $f(B,e^t,t)\leq 0$. We then proceed the proof by the induction on recursive calls. At the basis of the recursion, the returned solution is optimal (and thus also satisfies 2-approximation), since we have $\mathcal{X}=\emptyset$. Next, for each inductive step, if S' is a 2-approximation with respect to maximizing $\sum f_b(B,e^t,t)\cdot x_{B,e^t,t}$, S is also such a 2-approximation because we have $f_b(\tilde{B},\tilde{e^t},\tilde{t})=0$ according to Eq. (16). Furthermore, as *Lines* 10-14 in *Algorithm* 2 ensure that S is a semi-maximal solution, S is also a 2-approximation with respect to maximizing $\sum f_a(B,e^t,t)\cdot x_{B,e^t,t}$, based on *Theorem* 3. Therefore, according to *Theorem* 2, we prove that S is a 2-approximation with respect to maximizing $\sum f(B,e^t,t)\cdot x_{B,e^t,t}$. This in turn confirms that *Algorithm* 2 is a 2-approximation algorithm for the relaxed ILP. ■

B. Non-Recursive 2-Approximation Algorithm

Note that, recursive algorithms normally use more memory than iterative ones, while the memory resources on IPN nodes are usually limited and expensive. Hence, we convert *Algorithm* 2 into a stack-based iterative one, as shown in *Algorithm* 3. *Line* 1 initializes a temporary profit of each $x_{B,e^t,t}=1$ as $f'(B,e^t,t)=f(B,e^t,t)$ and allocates an empty stack $\mathcal{K}$. *Lines* 2-4 work similarly as *Lines* 1-3 in *Algorithm* 2 to improve the efficiency of subsequent steps.

Lines 5-16 shows the main iteration of our algorithm. In each iteration, we first select a variable from $\mathcal{X}$, mark it as

Algorithm 3: Iterative Approximation Algorithm

Input: $\mathcal{X}$, $f(B, e^t, t)$, $\mathcal{C}$
Output: S

1 initialize $f'(B, e^t, t) = f(B, e^t, t)$, and a stack $\mathcal{K} = \emptyset$;
2 **for** *each bundle* $B \in \mathcal{Q}_v$ **do**
3 | remove $x_{B,e^t,t}$ from $\mathcal{X}$ if $f(B, e^t, t) \leq 0$;
4 **end**
5 **while** $\mathcal{X} \neq \emptyset$ **do**
6 | select a variable from $\mathcal{X}$ as $x_{\tilde{B},\tilde{e^t},\tilde{t}}$;
7 | push $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ into $\mathcal{K}$ and set $\tilde{f} = f'(\tilde{B}, \tilde{e^t}, \tilde{t})$;
8 | **for** *each* $x_{B,e^t,t} \in \mathcal{X}$ **do**
9 | | **if** $x_{B,e^t,t} \in \mathcal{A}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right) \cup \mathcal{I}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)$ **then**
10 | | | $f'(B, e^t, t) = f'(B, e^t, t) - \tilde{f}$;
11 | | **end**
12 | | **if** $f'(B, e^t, t) \leq 0$ **then**
13 | | | remove $x_{B,e^t,t}$ from $\mathcal{X}$;
14 | | **end**
15 | **end**
16 **end**
17 **while** $\mathcal{K} \neq \emptyset$ **do**
18 | pop a variable $x_{B,e^t,t}$ from $\mathcal{K}$;
19 | **if** $S \cup \{x_{B,e^t,t}\}$ *is a feasible solution* **then**
20 | | $S = S \cup \{x_{B,e^t,t}\}$;
21 | **end**
22 **end**
23 **return** S;

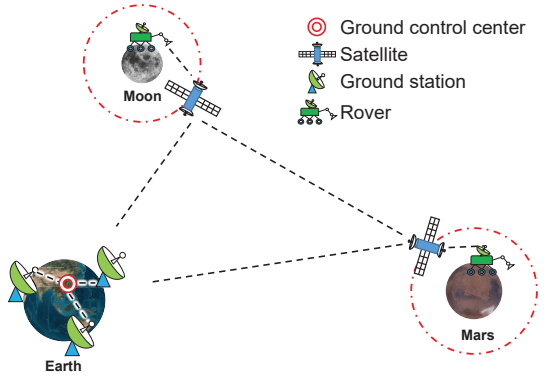


Fig. 4. Small-scale IPN topology (IPN-1).

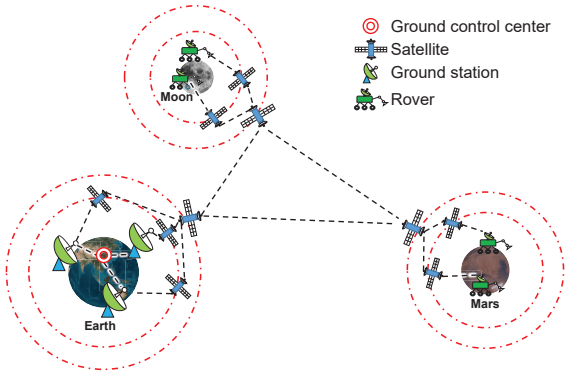


Fig. 5. Large-scale IPN topology (IPN-2).

$x_{\tilde{B},\tilde{e^t},\tilde{t}}$, push it into stack $\mathcal{K}$, and record the current temporary profit of $x_{\tilde{B},\tilde{e^t},\tilde{t}} = 1$ as $\tilde{f}$ (*Lines* 6-7). Then, *Lines* 8-15 update the temporary profit $f'(B, e^t, t)$ of each $x_{B,e^t,t} \in \mathcal{X}$, which is equivalent to calculating $f_b(B, e^t, t)$ in *Line* 8 of *Algorithm* 2, and delete a variable $x_{B,e^t,t}$ from $\mathcal{X}$ if its temporary profit becomes non-positive. This completes an iteration, and we iterate until $\mathcal{X}$ is empty. Note that, as $x_{\tilde{B},\tilde{e^t},\tilde{t}}$ itself belongs to $\mathcal{A}\left(x_{\tilde{B},\tilde{e^t},\tilde{t}}\right)$, it will be deleted from $\mathcal{X}$ in each iteration. Therefore, at least one variable will be removed from $\mathcal{X}$ in each iteration, verifying the convergence of *Algorithm* 3. Next, in *Lines* 17-22, we obtain the solution S by popping $x_{B,e^t,t}$ from stack $\mathcal{K}$ and adding it to S if feasible. This is equivalent to the recursive procedure in *Algorithm* 2.

Complexity Analysis: The major contributor to the time complexity of *Algorithm* 3 is the iterative procedure in *Lines* 5-16. Hence, if we define N as the largest-possible number of variables in $\mathcal{X}$ ($N = |\mathcal{Q}_v| \cdot |E_v^t| \cdot T_0$), the time complexity of *Algorithm* 3 is $O\left(N^2\right)$, indicating it is in polynomial time.

V. Performance Evaluations

In this section, we conduct simulations to evaluate our proposed FD-RDS algorithms (the ILP and the approximation algorithm) and compared them with existing benchmarks.

A. Simulation Setup

Our simulations consider two IPN topologies, which are a small-scale IPN (IPN-1) and a large-scale topology (IPN-2) in Figs. 4 and 5, respectively. IPN-1 consists of 8 nodes, including one ground control center, 3 ground stations, 2 rovers, and 2 satellites, where there are one ground control center, 3 ground stations, 4 rovers, and 10 satellites in IPN-2 (18 nodes). We leverage the Satellite Tool Kit (STK) [42] to emulate 24-hour node motions in the IPNs. Table II shows the simulation parameters. Except for those in Section V-E, all the simulations set the TS length and scheduling period length as $\Delta t = 1$ second and $T_0 = 64$ seconds, respectively.

On each node, the bundles are dynamically generated based on the Poisson traffic model. We assume that the bundles are in three priorities: high priority ($B(q) = 2$), middle priority($B(q) = 1$) and low priority ($B(q) = 0$). Meanwhile, we consider two traffic scenarios: 1) the uniform scenario where the source and destination, priority and size of each bundle are randomly chosen, and 2) the non-uniform scenario where bundles with different priorities have different distributions of source/destination nodes and sizes.

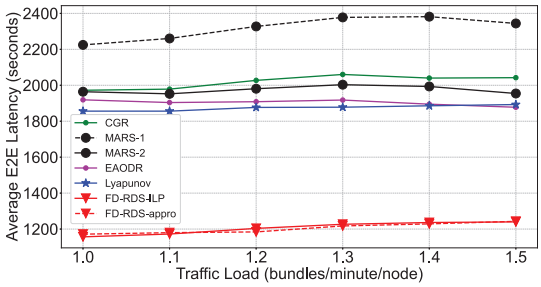
Specifically, in the non-uniform scenario, a high priority bundle contains control message sent by the ground control center to a satellite/rover with the size within $[1, 8]$ KByte, a middle priority bundle contains status information returned by a satellite/rover to the ground control center with the size within $[16, 64]$ KByte, and a low priority bundle contains scientific data for being transmitted in the IPN, where the ratios of the bundles for Earth-Moon, Earth-Mars and Moon-Mars are set as 20%, 40%, and 40%, respectively and the bundle size is within $[128, 1024]$ KByte. The ratio of the bundles in the three priorities is 1:1:2.

The simulations compare our ILP model (FD-RDS-ILP) and

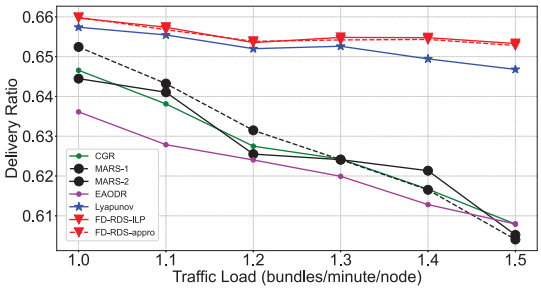
approximation algorithm in *Algorithm* 3 (FD-RDS-appro) with 4 benchmarks: CGR [17] and EAODR [22] (*i.e.*, the representative routing algorithms for IPNs), MARS-1 and MARS-2 (*i.e.*, the MARS in [33] with two weight settings, which are the pioneering approaches to address routing and data scheduling jointly in IPNs), and our Lyapunov optimization based approach in [34] (Lyapunov). Note that, as we let CGR first sort the bundles in a queue based on their priorities first and then process them sequentially, Overbooking Management [21] has actually been adopted in it. To make the performance comparisons realistic and fair, we include the running time that each algorithm takes to schedule each bundle on nodes in the E2E latency of the bundle. We average the results of 5 independent runs to get each data point in the simulations.

TABLE II
SIMULATION PARAMETERS

Parameters	Δt	T_0	average lifetime
Value	1 s	64 s	7200 s
uniform scenario	bundle size (KByte)		
	$[1, 1024]$		
non-uniform scenario	bundle size (KByte)		
	$B(q) = 2$	$B(q) = 1$	$B(q) = 0$
	$[1, 8]$	$[16, 64]$	$[128, 1024]$



(a) Average E2E latency



(b) Delivery ratio

Fig. 6. Uniform traffic scenario in IPN-1.

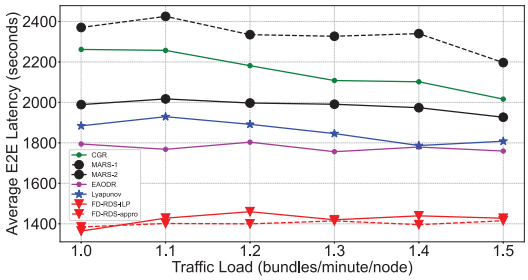
B. Simulations with Small-Scale IPN (IPN-1)

We first compare the performance of the algorithms in the small-scale IPN. Fig. 6 shows the results of the uniform traffic scenario. As expected, FD-RDS-ILP always outperforms all the benchmarks in terms of both average E2E latency and delivery ratio. Compared to the average performance of the benchmarks, FD-RDS-ILP and FD-RDS-appro (their average performance) achieve a reduction of 40.30% in average E2E latency and an improvement of 3.84% in delivery ratio. This is because our ILP for FD-RDS jointly considers a number of factors that can affect the performance of IP-DTs, including the projected delivery time, the latency of bundles sent over different paths, attributes of bundles, and contact plan of links, to obtain the optimal routing and data scheduling schemes on IPN nodes. We also observe that EAODR outperforms CGR in terms of average E2E latency, but its delivery ratio is slightly lower than that of CGR. Furthermore, we notice that Lyapunov outperforms other benchmarks, which confirms the importance of considering data scheduling in an IPN [34].

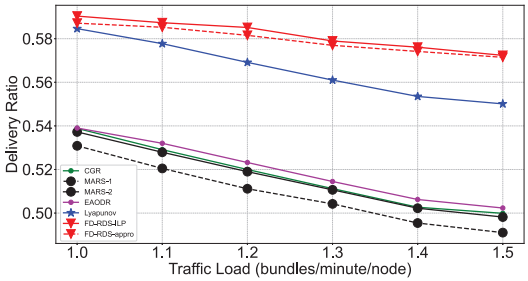
We can also see that the performance of the FD-RDS-appro is similar to that of FD-RDS-ILP, in terms of both average E2E latency and delivery ratio. Moreover, it is interesting to observe that the average E2E latency from FD-RDS-appro can even be slightly shorter than FD-RDS-ILP at certain traffic loads. This is because we take each algorithm's running time into account and add it to the E2E latency of each bundle. In fact, the running time of other algorithms is very short and thus can be ignored except for that of FD-RDS-ILP. This verifies the time-efficiency of our approximate algorithm (FD-RDS-appro), *i.e.*, FD-RDS-appro can balance the tradeoff between time-efficiency and FD-RDS performance better.

We then repeat the simulations with the non-uniform traffic scenario and plot the results in Fig. 7. Due to the larger portion of bundles with large size and long transmission distance, the IPN actually becomes more congested than that with the uniform traffic scenario. Therefore, all the algorithms perform worse in terms of both average E2E latency and delivery ratio, when the traffic load is the same. FD-RDS-ILP and FD-RDS-appro still always outperform the benchmarks in both average E2E latency and delivery ratio, achieving 30.10% reduction in average E2E latency and 10.49% improvement in delivery ratio, and the overall trend of the results is similar as that in Fig. 6. We can see that EAODR actually performs better in the non-uniform traffic scenario, as the average E2E latency from it can be even shorter than that from Lyapunov. Lyapunov still outperforms EAODR in terms of delivery ratio, even though the delivery ratio from EAODR is slightly higher than that from CGR this time. We also notice that the performance gaps between Lyapunov and FD-RDS-ILP and FD-RDS-appro are smaller. This is because the non-uniform traffic scenario can make the link usage in the IPN unbalanced, which will make the benefit of Lyapunov more obvious as it schedules bundles according to their outgoing links.

When comparing FD-RDS-ILP and FD-RDS-appro again, we observe that the advantage of FS-RDS-appro on average E2E latency becomes more obvious, due to the increased running time of FD-RDS-ILP to tackle a more congested IPN. As for the results on delivery ratio, the average gap between FS-RDS-appro and FD-RDS-ILP increases to 0.67%, which is 0.02% in the uniform traffic scenario, but FS-RDS-appro still outperforms all the benchmarks. Finally, we hope to point out that even though FD-RDS-ILP needs to solve the ILP, it can still run fast enough to schedule bundles timely. Specifically, for all the simulations discussed in this subsection, the average time that FD-RDS-ILP takes to schedule bundles on a node in a scheduling period is only 1.086 seconds, which is much shorter than a scheduling period (*i.e.*, $T_0 = 64$ seconds).

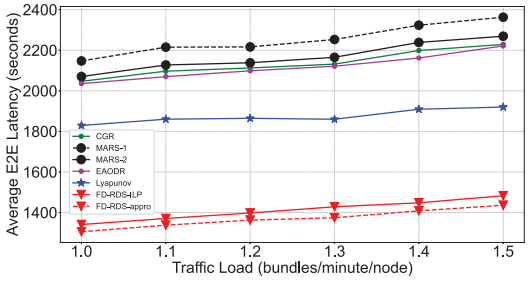


(a) Average E2E latency

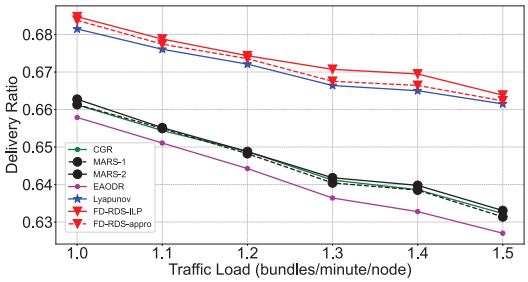


(b) Delivery ratio

Fig. 7. Non-uniform traffic scenario in IPN-1.



(a) Average E2E latency



(b) Delivery ratio

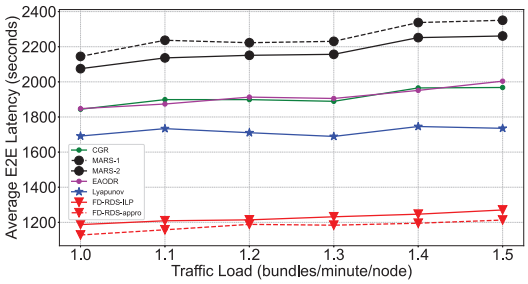
Fig. 8. Uniform traffic scenario in IPN-2.

C. Simulations with Large-Scale IPN (IPN-2)

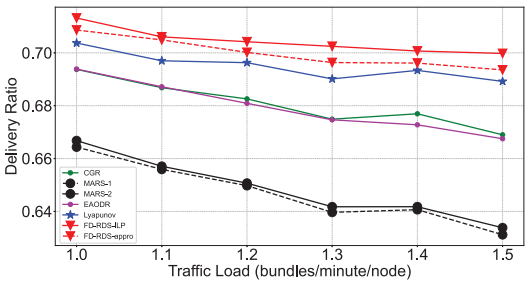
We repeat the simulations with the large-scale IPN, and the results are shown in Figs. 8-11. Among them, Figs. 8 and 10 show the average E2E latency and delivery ratio of each algorithm in uniform traffic and non-uniform traffic scenarios, respectively. As the connectivity among the nodes in IPN-2 is higher than that in IPN-1, it is easier for an algorithm to find E2E routing paths for bundles, and thus the delivery ratios of all the algorithms become higher. The algorithms perform similarly as those with IPN-1, and FD-RDS-ILP and FD-RDS-appro still always outperform all the benchmarks in terms of both average E2E latency and delivery ratio. In the two scenarios, compared with the benchmarks, our approaches on average achieve reductions of 34.06% and 38.18% in average E2E latency, and improvements of 3.47% and 6.86% in delivery ratio, respectively. This verifies that the advantages of our proposals over the benchmarks are not sensitive to the actual topology of an IPN or the attributes of bundles in it.

We notice that the average E2E latency of FD-RDS-appro is still shorter than that of FD-RDS-ILP, and the gap becomes larger compared with that in IPN-1. In addition to the longer running time of FD-RDS-ILP to tackle a larger-scale IPN, the larger gap can also be caused by the fact that bundles need to be transmitted over longer distances are more difficult to be delivered. Specifically, as shown in Figs. 8(b) and 10(b), the delivery ratio of FD-RDS-appro is slightly lower than that of FD-RDS-ILP, and most of the undelivered bundles actually have longer E2E latencies, which are not included in the calculation of the average E2E latency from FD-RDS-appro. This to some extent contributes to the shorter average E2E latency from FD-RDS-appro. The gaps on delivery ratio between FD-RDS-appro and FD-RDS-ILP are 0.27% and 0.95% in the uniform traffic and non-uniform traffic scenarios, respectively. Although the gaps are slightly larger than those observed in IPN-1, FD-RDS-appro still runs quickly with an average running time of 0.0032 seconds. The average running time of FD-RDS-ILP to schedule bundles in a scheduling period is 1.596 seconds, which is still much shorter than T_0.

Moreover, to further investigate the algorithms' performance on bundles in different priorities, we plot their performance on the bundles in the high priority in Figs. 9 and 11. We can see that compared with the results in Figs. 8 and 10, FD-RDS-ILP and FD-RDS-appro achieves larger performance gains over the benchmarks, achieving reductions of 39.71% and 46.56% in average E2E latency, and improvements of 4.78% and 4.53% in delivery ratio, respectively. We also find

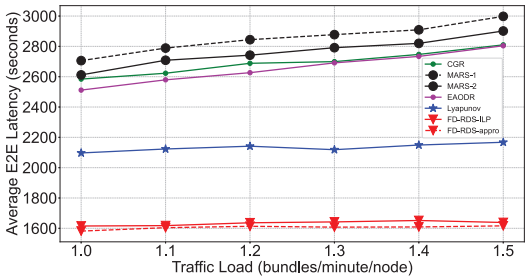


(a) Average E2E latency of the highest priority bundles

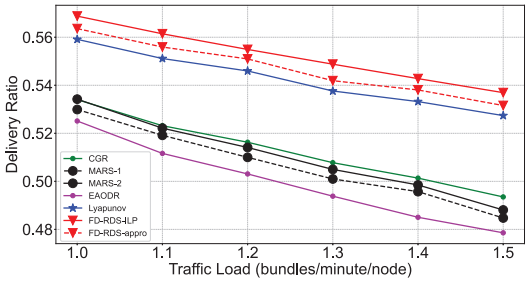


(b) Delivery ratio of the highest priority bundles

Fig. 9. High priority bundles of uniform traffic scenario in IPN-2.

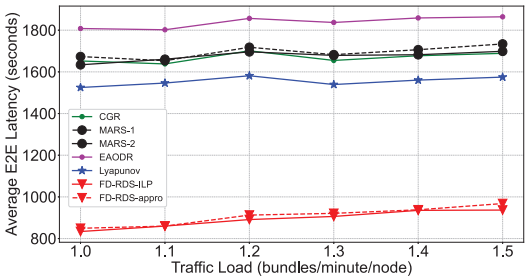


(a) Average E2E latency

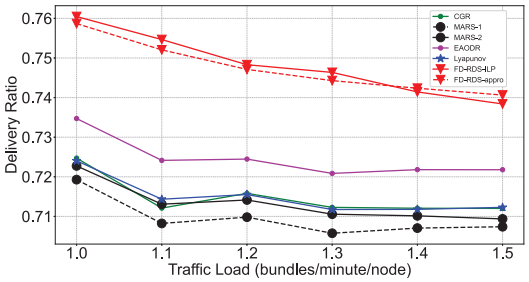


(b) Delivery ratio

Fig. 10. Non-uniform traffic scenario in IPN-2.



(a) Average E2E latency of the highest priority bundles



(b) Delivery ratio of the highest priority bundles

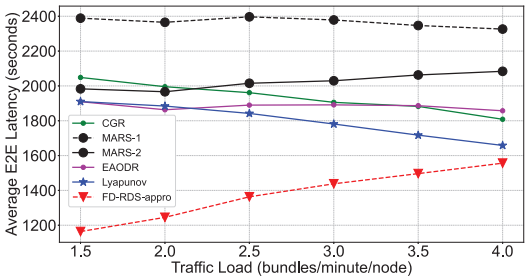
Fig. 11. High priority bundles of non-uniform traffic scenario in IPN-2.

that FD-RDS-appro and FD-RDS-ILP perform similarly when handling high-priority bundles. The average E2E latency of FD-RDS-appro remains close to or slightly lower than that of FD-RDS-ILP, and the gaps on delivery ratio between them are only 0.63% and 0.10% in the uniform traffic and non-uniform traffic scenarios, respectively. This suggests that although we relaxes the priority constraints before designing FD-RDS-appro, good performance can still be achieved for high-priority bundles. Note that, in the simulations, the benchmarks also process bundles based on their priorities, *i.e.*, CGR, EAODR and Lyapunov sort bundles in outgoing queues according to their priorities, while MARS-1 and MARS-2 take priority as an attribute to evaluate the utility of transmitting bundles. Fig. 11 also shows that EAODR provides a higher delivery ratio for high-priority bundles, but its average E2E latency is longer than other benchmarks. However, all the benchmarks cannot process bundles in different priorities as well as FD-RDS-ILP and FD-RDS-appro, as verified by the results in Figs. 8-11.

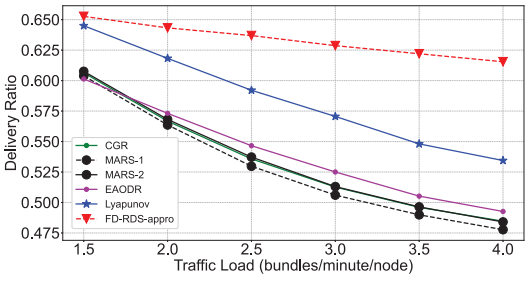
D. Simulations of Extremely High Traffic Loads

Next, we conduct simulations with extremely high traffic loads to evaluate the scalability of the algorithms. Specifically, we consider both IPN-1 and IPN-2 and set the traffic loads within $[1.5, 4.0]$ bundles/minute/node in the uniform traffic scenario. Note that, in this case, the running time of FD-RDS-ILP would become unacceptable, and thus we only compare FD-RDS-appro with the benchmarks.

The results for IPN-1 and IPN-2 are presented in Figs. 12 and 13, respectively. Fig. 12(b) shows that the delivery ratios from all the benchmarks decrease significantly. However, our FD-RDS-appro exhibits a slower rate of delivery ratio decline. In other words, as the traffic load increases, the advantage of FD-RDS-appro over the benchmarks becomes more prominent. In Fig. 12(a), the average E2E latency of the benchmarks



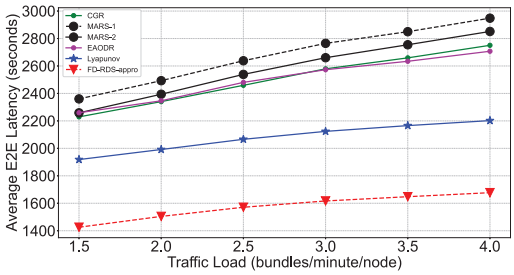
(a) Average E2E latency



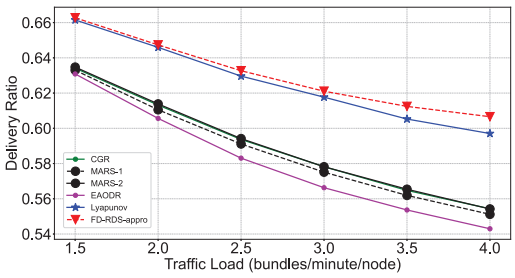
(b) Delivery ratio

Fig. 12. Extremely high traffic loads in IPN-1.

decreases with the increase of traffic load. As we previously analyzed, this is due to the decrease in delivery ratio, making it more difficult to deliver bundles with longer E2E latency. In contrast, our FD-RDS-appro strives to ensure the delivery of such bundles even as the traffic load increases, resulting in a gradual increase in average E2E latency that still remains shorter than that of other benchmarks. As depicted in Fig. 13, due to the better connectivity of IPN-2, the average E2E latency from the algorithms increases and the delivery ratio decreases with the traffic load. Nonetheless, our FD-RDS-appro still outperforms all the benchmarks and its advantage increases with the traffic load. Hence, even in extremely

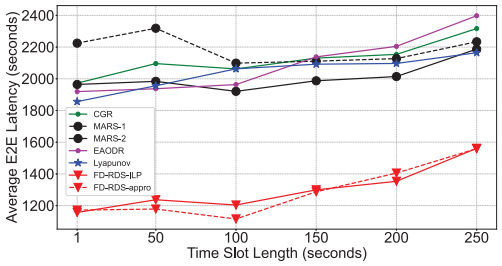


(a) Average E2E latency

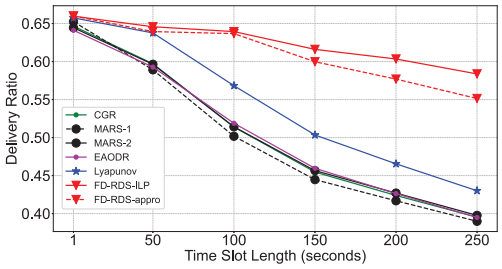


(b) Delivery ratio

Fig. 13. Extremely high traffic loads in IPN-2.



(a) Average E2E latency



(b) Delivery ratio

Fig. 14. Performance of IP-DTs with different TS lengths.

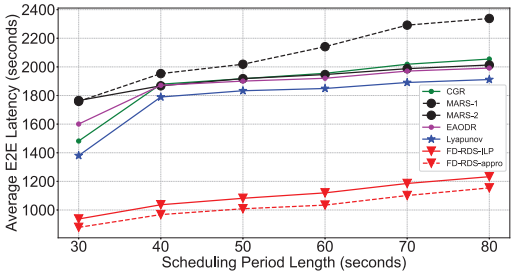
congested IPNs, our proposal can still achieve the best IP-DT performance. On average, compared with the benchmarks, FD-RDS-appro achieves reductions of 31.18% and 36.19% in average E2E latency, and improvements of 16.31% and 6.01% in delivery ratio in IPN-1 and IPN-2, respectively.

E. Sensitivity Analysis of TS and Scheduling Period Lengths

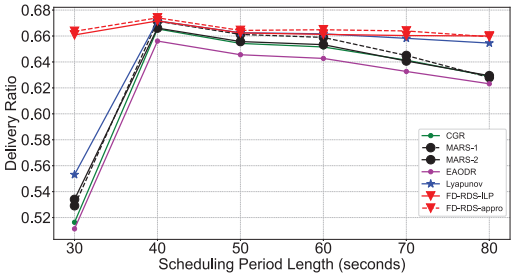
Finally, we perform simulations to analyze the algorithms' performance under different TS and scheduling period lengths. Here, the IPN topology is IPN-1, traffic scenario is the uniform scenario, and traffic load is 1 bundle/minute/node. This provides us insights on how to choose these two parameters empirically for an IPN. We have also tried other simulation scenarios and confirmed that similar trends can be seen.

Fig. 14 shows the impact of TS length Δt. We can see that the algorithms' performance degrades with the increases of Δt. This is because Δt intrinsically determines how fast the algorithms can react to network status changes. Meanwhile, using a Δt that is too short will trigger an algorithm too frequently, leading to unnecessary network status changes. Hence, we aim to choosing a Δt that can balance the two factors properly and $\Delta t = 1$ second is a reasonably choice.

The algorithms' performance with different scheduling period lengths (T_0) is shown in Fig. 15. As expected, the average E2E latency in Fig. 15(a) increases with T_0 for all the algorithms. As for the delivery ratio in Fig. 15(b), the algorithms' performance will not be good if T_0 is too short or too long. This is because if T_0 is too short, an algorithm for IP-DTs will only have very limited visibility of future information and thus can hardly optimize the routing and data scheduling of IP-DTs effectively. On the other hand, when T_0 is too long, the timeliness of the algorithm will be affected. Fig. 15 suggest that $T_0 = 40$ seconds is a reasonable setting since it strikes a balance between the visibility of future information and the timeliness of routing and data scheduling decisions.



(a) Average E2E latency



(b) Delivery ratio

Fig. 15. Performance of IP-DTs with different scheduling period lengths.

VI. Conclusion

In this work, we tried to optimize the routing and data scheduling of IP-DTs in an IPN. We formulated an ILP model and designed an effective distributed routing and data scheduling algorithm based on it (namely, FD-RDS-appro), which can realize bundle-level scheduling on each node in an IPN to maximize the delivery ratio of IP-DTs as well as minimizing their average E2E latency. We proved that FD-RDS-appro is a polynomial-time 2-approximation algorithm. Extensive simulations verified the performance of our proposals, indicating that they can effectively address the routing and data scheduling problem in an IPN. Specifically, our proposed algorithms outperformed known benchmarks in terms of both

the delivery ratio and E2E latency of IP-DTs, reducing the average E2E latency by $[30.10\%, 40.30\%]$ and improving the delivery ratio by $[3.47\%, 16.31\%]$ specifically.

REFERENCES

[1] "Cisco Annual Internet Report (2018-2023)." [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-papper-c11-741490.html.

[2] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[3] J. Liu *et al.*, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[4] C. Li *et al.*, "Overview of the Chang'e-4 mission: Opening the frontier of scientific exploration of the Lunar far side," *Space Sci. Rev.*, vol. 217, pp. 1–32, Jan. 2021.

[5] ——, "China's Mars exploration mission and science investigation," *Space Sci. Rev.*, vol. 217, pp. 1–24, May 2021.

[6] A. Alhilal, T. Braud, and P. Hui, "The sky is NOT the limit anymore: Future architecture of the interplanetary Internet," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, pp. 22–32, Aug. 2019.

[7] M. Marchese, "Interplanetary and pervasive communications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 26, pp. 12–18, Feb. 2011.

[8] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[9] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[10] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[11] M. Ju, F. Zhou, S. Xiao, and Z. Zhu, "Power-efficient protection with directed p-cycles for asymmetric traffic in elastic optical networks," *J. Lightw. Technol.*, vol. 34, pp. 4053–4065, Sept. 2016.

[12] C. Huang, X. Wang, and X. Wang, "Effective-capacity-based resource allocation for end-to-end multi-connectivity in 5G IAB networks," *IEEE Trans. Wirel. Commun.*, vol. 21, pp. 6302–6316, Aug. 2022.

[13] S. Li, D. Hu, W. Fang, and Z. Zhu, "Source routing with protocol-oblivious forwarding (POF) to enable efficient e-health data transfers," in *Proc. of ICC 2016*, pp. 1–6, Jun. 2016.

[14] C. Chen *et al.*, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 4701–4711, Dec. 2014.

[15] S. Li *et al.*, "Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability," *IEEE Netw.*, vol. 31, pp. 58–66, Mar./Apr. 2017.

[16] J. Mukherjee and B. Ramamurthy, "Communication technologies and architectures for space network and interplanetary Internet," *IEEE Commun. Surveys Tuts.*, vol. 15, pp. 881–897, Second Quarter 2013.

[17] G. Araniti *et al.*, "Contact graph routing in DTN space networks: overview, enhancements and performance," *IEEE Commun. Mag.*, vol. 53, pp. 38–46, Mar. 2015.

[18] N. Alessi *et al.*, "DTN performance in complex deep-space networks," in *Proc. of ASMS/SPSC 2018*, pp. 1–7, Sept. 2018.

[19] E. Birrane, S. Burleigh, and N. Kasch, "Analysis of the contact graph routing algorithm: Bounding interplanetary paths," *Acta Astronaut.*, vol. 75, pp. 108–119, Jun./Jul. 2012.

[20] N. Bezirgiannidis, F. Tsapeli, S. Diamantopoulos, and V. Tsaoussidis, "Towards flexibility and accuracy in space DTN communications," in *Proc. of ACM CHANTS 2013*, pp. 43–48, Sept. 2013.

[21] N. Bezirgiannidis *et al.*, "Contact graph routing enhancements for delay tolerant space communications," in *Proc. of ASMS/SPSC 2014*, pp. 17–23, Sept. 2014.

[22] S. El Alaoui and B. Ramamurthy, "EAODR: A novel routing algorithm based on the modified temporal graph network model for DTN-based interplanetary networks," *Comput. Netw.*, vol. 129, pp. 129–141, Dec. 2017.

[23] S. Burleigh *et al.*, "Delay-tolerant networking: an approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, pp. 128–136, Jun. 2003.

[24] K. Scott and S. Burleigh, "Bundle protocol specification," *RFC 5050*, Nov. 2007. [Online]. Available: http://tools.ietf.org/html/rfc5050.

[25] V. Cerf *et al.*, "Delay-tolerant networking architecture," *RFC 4838*, Apr. 2007. [Online]. Available: https://tools.ietf.org/html/rfc4838.

[26] S. Burleigh, "Interplanetary overlay network: An implementation of the DTN bundle protocol," in *Proc. of CCNC 2007*, pp. 222–226, Jan. 2007.

[27] J. Mukherjee and B. Ramamurthy, "The interplanetary internet implemented on a terrestrial testbed," *Ad Hoc Netw.*, vol. 27, pp. 147–158, Jan. 2015.

[28] J. Fraire, O. De Jonckère, and S. Burleigh, "Routing in the space Internet: A contact graph routing tutorial," *J. Netw. Comput. Appl.*, vol. 174, p. 102884, Jan. 2021.

[29] F. De Rango and M. Tropea, "DTN architecture with resource-aware rate adaptation for multiple bundle transmission in interplanetary networks," *IEEE Access*, vol. 10, pp. 47 219–47 234, Apr. 2022.

[30] P. Lu, Q. Sun, K. Wu, and Z. Zhu, "Distributed online hybrid cloud management for profit-driven multimedia cloud computing," *IEEE Trans. Multimedia*, vol. 17, pp. 1297–1308, Aug. 2015.

[31] K. Wu, P. Lu, and Z. Zhu, "Distributed online scheduling and routing of multicast-oriented tasks for profit-driven cloud computing," *IEEE Commun. Lett.*, vol. 20, pp. 684–687, Apr. 2016.

[32] X. Xie *et al.*, "Evacuate before too late: Distributed backup in inter-DC networks with progressive disasters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, pp. 1058–1074, May 2017.

[33] S. El Alaoui and B. Ramamurthy, "MARS: A multi-attribute routing and scheduling algorithm for DTN interplanetary networks," *IEEE/ACM Trans. Netw.*, vol. 28, pp. 2065–2076, Oct. 2020.

[34] X. Tian and Z. Zhu, "On the distributed routing and data scheduling in interplanetary networks," in *Proc. of ICC 2022*, pp. 1109–1114, May 2022.

[35] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan and Claypool, 2010.

[36] C. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[37] M. Goemans and D. Williamson, *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., 1996.

[38] A. Bar-Noy *et al.*, "A unified approach to approximating resource allocation and scheduling," *J. ACM.*, vol. 48, pp. 1069–1090, Sept. 2001.

[39] M. Cheung, J. Mestre, D. Shmoys, and J. Verschae, "A primal-dual approximation algorithm for min-sum single-machine scheduling problems," *SIAM J. Discret. Math.*, vol. 31, pp. 825–838, May 2017.

[40] M. Gary and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co. New York, 1979.

[41] R. Bar-Yehuda and S. Even, "A local-ratio theorem for approximating the weighted vertex cover problem," *North-Holland Math. Stud.*, vol. 109, pp. 27–45, Dec. 1985.

[42] Satellite tool kit. [Online]. Available: http://www.agi.com/products/stk/.