Routing Heterogeneous Traffic in Delay-Tolerant Satellite Networks

Pablo G. Madoery, Member, IEEE, Güneş Karabulut Kurt, Senior Member, IEEE, Halim Yanikomeroglu, Fellow, IEEE, Peng Hu, Senior Member, IEEE, Khaled Ahmed, Stéphane Martel, and Guillaume Lamontagne

Abstract—Delay-tolerant networking (DTN) offers a novel architecture that can be used to enhance store-carry-forward routing in satellite networks. Since these networks can take advantage of scheduled contact plans, distributed algorithms like the Contact Graph Routing (CGR) can be utilized to optimize data delivery performance. However, despite the numerous improvements made to CGR, there is a lack of proposals to prioritize traffic with distinct quality of service (QoS) requirements. This study presents adaptations to CGR to improve QoS-compliant delivery ratio when transmitting traffic with different latency constraints, along with an integer linear programming optimization model that serves as a performance upper bound. The extensive results obtained by simulating different scenarios show that the proposed algorithms can effectively improve the delivery ratio and energy efficiency while meeting latency constraints.

Index Terms—Routing, delay-tolerant networks, satellite constellations, QoS, contact graph routing, congestion avoidance.

I. INTRODUCTION

The satellite industry has experienced unprecedented advances in microelectronics, commercial off-the-shelf solutions, and rocket launch platforms that have reduced the cost of manufacturing and deploying satellites [1]. The total number of active satellites orbiting the Earth has increased from 1,459 in 2016 to 4,852 in 2022 and is planned to reach 100,000 satellites by 2030. In addition, the possibility of using inter-satellite links and on-board processing capabilities turns these satellite constellations into shared resource networks capable of providing ubiquitous sensing, navigation, positioning, and communication services.

The provision of this great diversity of services, with a heterogeneous network structure, reveals the need to use innovative communication protocols that can cope with variable delays and losses, as well as link disruptions caused by the high mobility of satellites with respect to terrestrial users and other satellites.

In this regard, delay-tolerant networking (DTN) has been identified as a novel approach that considers delays and disruptions not as failures but as characteristics of this type of networks [2]. Therefore, the proposed architecture allows addressing these challenges from the design of the communication protocols and without requiring an excessive network infrastructure, such as the current satellite mega-constellations that aim to completely avoid such interruptions or delays. The fundamental part of the architecture is the Bundle Protocol (BP) that exists at a layer above the transport layers of the network. BP proposes the use of persistent storage on each DTN node to store-carry-and-forward data packets as transmission opportunities become available.

In the case of satellite networks, the subsequent communication episodes can be determined in advance based on orbital dynamics. These types of deterministic DTNs are referred to as scheduled DTNs, and can leverage a contact plan that comprises the future network topology in order to optimize routing. The contact graph routing (CGR) algorithm, described in [3], is the most mature development for these networks and has been the focus of extensive contributions from the research community. These include the adaptation of the Dijkstra shortest path algorithm to time-varying networks [4], the prevention of routing loops and consideration of multiple destinations [5], [6], source routing extensions [7], and the application of overbooking management [8], [9]. In addition to this, other improvements include congestion mitigation techniques [10], route table management strategies and the incorporation of Yen’s algorithm [11]. Furthermore, the incorporation of opportunistic [12] and probabilistic contacts [13], the design of new schemes to enhance the scalability [14], a spanning-tree formulation to compute routes to several destinations [15], and a partial queue information sharing [16], as well as many other contributions.

Additionally, recent works have begun to incorporate the required quality of service (QoS) as an important attribute to be taken into account in routing [17], [18], [19]. In
particular, [17] proposes a centralized approach that involves the creation of a storage time aggregated graph (STAG) to represent the time-varying topology, followed by the assignment of mission’s data to specific routes in the STAG. Each mission has different latency requirements and the objective is to maximize the number of missions delivered to the destination. Using a centralized scheme has some advantages such as the potential to leverage global knowledge of both topology and traffic to derive more efficient solutions. On the other hand, centralized schemes suffer from some disadvantages such as overhead of communication with a central entity, increased complexity, lower scalability, single point of failure problem, reduced capability to react to failures, etc.

Nevertheless, although the community has shown considerable interest, no substantial proposals have been made to adapt a distributed algorithm such as CGR to prioritize traffic with different QoS requirements. Hence, this paper has a twofold contribution: first, we suggest modifications to the route selection process to prioritize traffic based on the required QoS. Second, we model the problem using a integer linear programming optimization model that provides a performance upper bound. This paper is an extended version of [20] and includes additional contributions such as an improved description of the CGR-Hops routing scheme, a new multi-objective routing scheme called CGR-MO, and new performance metrics for evaluation. In addition, we have extensively evaluated a realistic Walker-Delta constellation scenario and our numerical results demonstrate improvements in delivery ratio and energy efficiency while meeting maximum delay requirements.

It is worth noting that the scope of these contributions is framed within sparse and mid-size satellite networks that are well suited to be modeled with the DTN paradigm, which targets networks with high propagation delays and/or disruptions [2]. Most of the time, the nodes of these networks do not have end-to-end connectivity through multi-hop routes. Therefore, they must be able to buffer traffic when no communication is possible, perhaps for long periods of time, and send the traffic when communication becomes available. Although mega-constellations of satellites could take advantage of the proposed routing schemes to forward delay tolerant traffic, this case is beyond the scope of this paper.

The remainder of the paper is organized as follows. In Section II, we describe the processes involved in routing in scheduled satellite networks, including a description of the CGR algorithm and proposed adaptations to account for QoS requirements. Additionally, we provide an optimal traffic flow model together with an execution example comparing the behavior of the routing schemes in a specific scenario. Next, in Section III we describe the simulation environment, two different simulated scenarios, the performance metrics used and the results obtained. Finally, we describe future lines of research in Section IV, and conclude the work in Section V.

II. ROUTING SCHEMES

A. Routing in Scheduled Satellite Networks

Four typical processes are involved in transmitting traffic from origins to destinations using scheduled communications within delay-tolerant satellite networks:

- **Plan**: the determination of contact plans relies on a central entity, such as a ground station or mission control center, which estimates future communication episodes. Each episode of communication between 2 nodes is known as a contact and has a start time, an end time, and an associated data rate. To complete this task, it is necessary to consider the physical arrangement and orientation of nodes over time, along with their communication system configuration, which includes aspects such as antenna, transponder, modulation and transmission power. The combination of orbital propagators and communication models yields a contact plan that can be adjusted to minimize energy consumption or eliminate conflicting contacts [21], [22], [23]. Once the contact plan is generated, it is disseminated to the satellites, which can then execute distributed routing processes when traffic is generated or received by different nodes within the network. Although the design of the contact plan may have an important effect on the QoS obtained, this paper focuses on routing and therefore considers scenarios where the contact plan is trivial or is already defined.

- **Routing**: in contrast to stable networks like the Internet, where routes are established as a series of nodes, each satellite in a delay-tolerant satellite network builds a routing table using the contact plan as an input. The routes in these networks are constructed as a sequence of temporary contacts and have associated time periods during which they are valid (they become invalid when one of their contacts terminates). Various attributes, such as estimated delivery time, number of hops/contacts, and remaining capacity, can be calculated for these routes.

- **Forwarding**: If a satellite generates or receives traffic, it consults its routing table, filters out routes that are no longer valid, and selects the one that optimizes a metric, typically delivery time. Using the information linked to the selected route, the satellite places the traffic in a queue that corresponds to a specific neighboring node (also called the next hop).

- **Transmission**: Finally, when contacts are established between satellites, the enqueued traffic is transmitted in a specific order.

B. Contact Graph Routing (CGR)

As detailed in [3], CGR is a routing algorithm that operates in a distributed fashion on several satellites. The algorithm’s first task is to build a routing table (routing process), which is followed by the selection of routes for enqueuing traffic (forwarding process). To construct the routing table, the approach utilizes the contact plan as input and constructs a contact graph structure, where vertices correspond to contacts and arcs represent data retention events in a satellite. Although this static structure may not be immediately intuitive, it simplifies the execution of route search algorithms in a topology that varies over time. The search process utilizes the contact graph structure and is an adapted version of Dijkstra’s algorithm that explores the different contacts (nodes in the contact graph) and selects those that determine the optimal route in terms of earliest delivery.
time to the destination. To compare it with other schemes, we will refer to this scheme as \textit{CGR-DelTime}. Moreover, CGR computes not just one route to a destination but instead the \( K \) best routes using Lawler’s modification of Yen’s algorithm. This is due to the fact that routes have a deadline and limited capacity.

Subsequently, in the second stage, during the forwarding process, CGR begins with a packet and the routing table constructed in the preceding stage. It filters out routes that are either no longer viable or have reached their capacity and then chooses the most suitable route based on the earliest delivery time metric. Following this, the process is reiterated for every new packet that needs forwarding, with annotations indicating the capacity used from the route.

It is important to note that CGR considers the time-to-live (TTL) of a packet during the process of filtering routes. It disregards routes that have a projected delivery time later than the required time. However, selecting a route based solely on estimated delivery time could result in a situation where the required time. However, selecting a route based solely on estimated delivery time could result in a situation where the effective delivery time and the estimated delivery time is a metric computed locally where downstream nodes cannot honor the selected route, and the traffic fails to reach the destination within the required timeframe.

It is, therefore, useful to make a clear distinction between the estimated delivery time and the effective delivery time. The estimated delivery time is a metric computed \textit{locally} at each node, and in the case of CGR it only takes into account topological aspects, i.e., what is the starting time of the last contact of a route. On the other hand, the effective delivery time is the time in which a packet is effectively delivered to its destination. This time may be later than the estimated delivery time due to traffic generated by intermediate nodes or unforeseen global network conditions. In general, the greater the number of hops in the chosen route, the greater the difference between the effective delivery time and the estimated delivery time.

C. Proposed Contact Graph Routing Adaptations

1) \textit{CGR-Hops}: To address the above-mentioned problems, this work considers a variation of CGR in the forwarding stage. This new scheme called \textit{CGR-Hops} will still filter out those routes that do not deliver a packet in the required time, like the original CGR-DelTime scheme, but will then choose the best route as the one with the lowest number of hops.

Therefore, CGR-Hops does not propose simply changing the delivery time metric by the number of hops in all the processes of CGR. The first filtering stage that \textit{CGR-DelTime} performs to let through only those routes whose estimated delivery time is less than the TTL is maintained. It is after this process that CGR-Hops chooses, among the routes that passed the filtering, the one with the lowest number of hops. This allows, on the one hand, to take advantage of the locally estimated delivery time of a route and on the other hand, not to choose routes with fewer hops but with late delivery times.

It is anticipated that this approach will take decisions that ease congestion by utilizing a smaller number of hops. This reduction in congestion is expected to free up capacity for traffic that requires earlier arrival.

2) \textit{CGR-MO}: To explore the selection of routes with intermediate values of the number of hops and estimated delivery time, we propose a multi-objective adaptation of CGR called \textit{CGR-MO}. \textit{CGR-MO} maintains the same routing and route filtering stages as \textit{CGR-DelTime} and \textit{CGR-Hops}. The difference is during the forwarding stage and after route filtering. \textit{CGR-MO} calculates a metric \( M \) associated with each of the computed routes and then chooses the route with the lowest value of \( M \). \( M \) is defined according to Equation (1), where \( w \) is a weight that can vary in the range \([0, 1]\), and where \( \text{norm}_\text{hops} \) and \( \text{norm}_\text{delivery\_time} \) are attributes of each route but normalized using the information present in the contact plan. In particular, \( \text{norm}_\text{hops} \) is computed as the \( \text{hops\_number} \) of the route divided by the total number of nodes in the network, while \( \text{norm}_\text{delivery\_time} \) is computed as the \( \text{estimated\_delivery\_time} \) of a route divided by the contact plan time horizon. Therefore, the values of \( w \) closer to 1 will give more importance to the number of hops, providing a behavior closer to \textit{CGR-Hops}, while values of \( w \) closer to 0 will give more importance to the delivery time, providing a behavior closer to \textit{CGR-DelTime}.

\[
M = w \times \text{norm}_\text{hops} + (1 - w) \times \text{norm}_\text{delivery\_time} \tag{1}
\]

D. Optimal Traffic Flow Integer Linear Programming Model

To assess various routing strategies, it is beneficial to count with an optimization model. Although it may not be feasible to apply this model in practical networks due to computational complexity, it can still be utilized as an upper limit of performance to evaluate the proposed techniques in smaller networks.

The literature refers to the challenge of optimally routing multiple traffic as “multi-commodity flow” [24]. Several studies, such as [17], [25], [26], [27], have presented integer linear programming models that can solve this problem in networks with intermittent and scheduled communications, specifically in the context of DTN networks. In this paper, we introduce a modified version of these models, customized to our particular requirements, and extended to accommodate traffic with varying quality of service in terms of latency.

1) Coefficients and Decision Variables: The network topology is divided into \( K \) states, where each state \( k_q \in K \) represents the network during a specific time interval \([t_{q-1}, t_q] \). Therefore, the network is composed of \( k_1, \ldots, k_q, \ldots, k_f \) states, which correspond to the time intervals \([t_0, t_1], \ldots, [t_{q-1}, t_q], \ldots, [t_{f-1}, t_f] \). The graph \( G_{k_q} \) represents the network in each state \( k_q \), where the vertices \( v \in V \) are nodes, and the arcs \( e \in \mathbb{E}_{k_q} \) are communication opportunities (contacts) between nodes in \( k_q \). The notation \( c_{e} \) represents the capacity, measured in the number of packets, that can be transmitted from the source node to the destination node of an arc \( e \) during one state \( k_q \). Additionally, the number of packets that can be stored in the buffer of node \( v \) is represented as \( b_{v} \).

For a more concise notation, we will use \( I^v \) to denote the set of arcs \( e \) entering a node \( v \), and \( O^v \) to denote the set of
TABLE I
OPTIMAL TRAFFIC FLOW MODEL PARAMETERS

| Input Coefficients | Timesteps |
|--------------------|-----------|
| $t_k \in T$        | $k_q = [t_{q-1}, t_q) \in K$ |
| $v \in V$          | $v \in V$ |
| $e \in E_{k_q}$    | $E_{k_q}$ |
| $c_e$              | Capacity of arc $e$ |
| $b_e$              | Capacity of buffer's node $e$ |

| Output Variables |
|------------------|
| $d_{v,TTL} \in D$| Traffic from node $y$ to node $x$ originated at timestamp $t_q$ with time-to-live TTL |

For arcs leaving. Moreover, $D$ represents the set of all the traffic demands $d_{v,TTL}$ generated at time $t_q$ from node $y$ to node $z$ with a time-to-live of TTL. We assume that all traffic demands are generated at the beginning of a state, but if this is not the case, it is possible to add an intermediate state to account for such traffic generation.

Regarding the output variables, $X_{y,z}^{x,v}$ denotes the number of packets of traffic $y$, $z$ sent on arc $e$, while $B_{v,TTL}^{x,v}$ represents the number of packets of traffic $y$, $z$ stored at node $v$ at time $t_q$. Table I summarizes these model parameters.

2) Objective Function and Constraints: Taking into account the coefficients and decision variables defined, we link them by means of the objective function defined in (2) subject to the constraints (3) to (8).

Thus, an integer linear programming (ILP) model is formulated with inputs comprising a set of traffic parameters ($d_{v,TTL}$) to be transmitted from sources ($y$) to destinations ($z$) through a time-varying topology with buffer capacities ($b_e$) and transmission capacities ($c_e$), and outputs consisting of optimal flows ($X_{y,z}^{x,v}$) to be followed by the traffic.

The objective function described in equation (2) aims to minimize the sum of products $w(k_q) \times X_{y,z}^{x,v}$, where $w(k_q)$ is a weighting function that assigns a higher weight to each state in increasing order. Therefore, minimizing this product prioritizes the delivery of traffic as soon as possible while minimizing the use of later arcs.

Several constraints are imposed in the model to ensure a consistent transmission of traffic. Equation (3) defines the buffer occupancy of each node ($v$) at each timestamp ($t_q$) using the previous state ($t_{q-1}$), the incoming ($e \in I^v$) and outgoing ($e \in O^v$) flows in the interval $[t_{q-1}, t_q)$, and the traffic generated at $t_q$ ($d_{v,TTL}$). This equation must hold for all $t_q, v, y, z$. Equation (4) limits the buffer occupancy ($B_{v,TTL}^{x,v}$) of each node to $b_v$, while equation (5) limits the traffic flow through each arc ($X_{y,z}^{x,v}$) to $c_e$.

Moreover, constraint (6) establishes the buffers at the initial timestamp ($t_0$), and constraint (7) ensures that traffic generated in intermediate states is not sent in earlier states. This constraint also guarantees that the traffic that has a maximum latency constraint (TTL) arrives at its destination on time and remains there until the final state. Finally, equation (8) ensures that all traffic generated over time reside in their respective buffers at the destination nodes ($B_{v,TTL}^{x,v}$) at timestamp $t_f$.

Table I summarizes these model parameters.

Subject to:

$$B_{v,TTL}^{x,v} = \begin{cases} d_{v,TTL}^{x,v} & \text{if } y = v \\ 0 & \text{if } y \neq v \end{cases} \quad \forall v, y, z \quad (6)$$

$$B_{v,TTL}^{x,v} = \begin{cases} d_{v,TTL}^{x,v} & \text{if } y = v \\ 0 & \text{if } y \neq v \end{cases} \quad \forall v, y, z \quad (7)$$

$$B_{v,TTL}^{x,v} = \begin{cases} \sum_{e \in T} d_{v,TTL}^{x,v} & \text{if } y = v \\ 0 & \text{if } y \neq v \end{cases} \quad \forall v, y, z \quad (8)$$

E. Execution Example

To gain an intuitive understanding of the routing schemes, we examine a basic example where three nodes (N1, N2, N3) each have 3 contacts in different states as illustrated in Figure 1(a). The dotted lines show contacts with the capacity to send 10 packets each. Nodes N1 and N2 both produce 10 packets in state k1 headed for N3 (containing in colored squares). N1’s traffic has a TTL of 30 seconds, so it must arrive at its destination by state k3 or earlier, while N2’s traffic has a TTL of 20 seconds, requiring it to reach the destination by state k2 or earlier. N1 can access routes R1 and R3 to reach N3, while N2 can use route R2, as depicted by solid blue lines.

The behavior of CGR-DelTime is shown in Figure 1(b), where colored solid lines depict traffic transmissions. N1 calculates and sends traffic using route R1, but when the packets reach N2, they encounter an unexpected situation: N2 also has traffic destined for N3 and must use the entire capacity of the N2-N3 contact. As a result, congestion arises, preventing N1’s 10 packets from reaching N3 within the required time. Conversely, N2’s traffic reaches N3 on time by following route R2.

Finally, Figure 1(c) displays the behavior obtained using both CGR-Hops and the ILP model. Although CGR-Hops knows that route R1 delivers the packets sooner, it chooses R3, which not only delivers the packets on time (TTL = 30 s) but also uses fewer hops, resulting in congestion avoidance. This effect is achieved by both the ILP model and CGR-Hops, resulting in more QoS-compliant traffic flows, and enabling all traffic to reach its destination on time. However, the ILP model requires global knowledge of the topology and traffic generated, whereas in this case CGR-Hops can achieve similar outcomes without knowing the traffic generated by other network nodes. To assess the potential of this approach, we
III. PERFORMANCE EVALUATION

A. Simulation Environment

To compare the performance of the routing schemes described in Section II, we have adapted and extended Dtnsim, a discrete event-driven simulator developed in the Omnet++ framework. DtnSim was originally presented in [28] and is available as open-source software in a public repository.\footnote{The public DtnSim repository can be found at: https://bitbucket.org/lcd-unc-ar/dtnsim}

As shown in Figure 2, each node (satellite or ground station) is modeled as an Omnet module with 3 internal sub-modules:

- **app**: in charge of generating and/or receiving traffic packets at a given time, with a source, destination, a size in bytes, and a TTL. This sub-module communicates directly with the **dm** sub-module.
- **dm**: in charge of storing, routing, and forwarding traffic packets. This sub-module implements all the routing schemes described and communicates directly with the **app** and **com** sub-modules.
- **com**: in charge of transmitting/receiving packets to/from other nodes. This sub-module communicates directly with the **dm** sub-module and possibly with the **com** sub-module of other nodes.

In addition to these sub-modules, a **central** module is responsible for reading a file containing the contact plan and distributing that information to all nodes of the network. Then, each node establishes contacts with its neighbors at the corresponding times. In this way, although the nodes remain visually static, the time-varying topology is captured and implemented through the input contact plan.

B. Simulation Scenarios

As described below, we performed simulations of 2 types of scenarios: Random Networks and Walker-Delta Constellation:

1) Random Networks: We generated 100 random network topologies of 100s divided into 10 states of 10s. An example topology can be seen in Figure 3. Each of these topologies model the time-varying connectivity among 11 nodes, with nodes 1-10 representing satellites and node 11 representing a ground station.

The connectivity between nodes is governed by a contact density parameter $\delta$ that ranges from 0.0 to 1.0. A network with $\delta = 1.0$ is fully connected with contacts present in all states, while $\delta = 0$ implies no contacts exist. For our study, we set $\delta = 0.2$. Increasing $\delta$ requires higher traffic generation to observe the same congestion and performance degradation. Each link has a capacity to send 10 traffic packets, and nodes have unlimited storage capacity.

To induce congestion, we used an all-to-one traffic pattern in which all 10 nodes generate variable traffic load directed to node 11. Nodes 1-5 generate packets without maximum latency requirements ($\text{TTL} \rightarrow \infty$), allowing them to reach
Fig. 3. Example of a random network consisting of 10 states.

Fig. 4. Walker-Delta constellation represented in 2 and 3 dimensions together with simulation parameters. Figure adapted from [31].

their destination in any state. In contrast, nodes 6-10 generate packets with a maximum latency requirement of 20 seconds, forcing them to arrive at their destination in either the first state \(k_1\) or the second state \(k_2\).

These parameters were chosen to ensure that heterogeneous traffic was routed, congestion was eventually provoked, and packets had a chance to arrive at their destination before the simulation ended.

An important clarification is that the random network generation consisted of two stages. A first stage in which 100 random networks were generated without any constraints, followed by a second stage in which only those networks were kept in which the ILP model was able to find an optimal routing solution for the maximum traffic load generated. After these two stages, 25 random networks were obtained that comply with the described property.

2) Walker-Delta Constellation: This type of constellation was first proposed in [29], studied in several works such as [30], [31], [32], and is widely used by real deployments such as the Starlink constellation operated by SpaceX.

For this work, we have used a sparser Walker-Delta constellation with the orbital and simulation parameters described in [31], and represented in Figure 4. This particular type of network is one of the so-called Ring Road Networks (RRNs) [33] and its purpose is to connect remote sites without Internet access (cold spots) with sites in urban regions with Internet access (hot spots) by means of small and inexpensive satellites that act as data mules to exchange data between hot spots and cold spots.

The constellation was propagated over a time horizon of 1 day and is composed of 4 orbital planes with 4 satellites per orbital plane. In addition, there are 25 ground targets (nodes 7 to 31), 6 ground stations (nodes 1 to 6), and a mission control center (MCC, node 48). The MCC is connected via the Internet to each ground station and satellites are capable of exchanging data through inter-satellite links (ISLs) that are established when they have line of sight and the distance between them is less than 1,000 km. Only 1 ISL needs to be established at the same time. Furthermore, satellites are capable of establishing links with ground stations (GSLs) when the same conditions are met. The resulting contact plan used for the simulations can be found in https://upcn.eu/icc2017.html.

With respect to traffic, each of the ground targets generates a variable traffic load destined to the MCC, while the MCC generates a variable traffic load destined to each of the ground targets. Half of the generated traffic packets have a TTL of 1 hour, while the other half has no TTL restriction (TTL → ∞).

C. Performance Metrics

We compare the performance of CGR-DelTime, CGR-Hops, CGR-MO with weights 0.25, 0.5, 0.75, and the ILP model described in Section II. We will be examining the following metrics in our analysis:

- **Delivery ratio**: the total number of generated packets divided by the total number of packets that arrive at their destination within their specified maximum latency.
- **Dropped packets per node**: the total number of packets dropped at each node. If a node’s routing scheme cannot find a route that provides a chance of meeting the specified maximum latency, the packet is discarded. This increases the value of this metric and decreases the value of the delivery ratio metric.
- **Mean hops per packet**: the total number of transmissions of all packets divided by the total number of packets that arrive at their destination within their specified maximum latency.
- **Energy efficiency**: the total number of packets that arrive at their destination within their specified maximum latency divided by the total number of transmissions of all packets.
Mean delay per packet: the average latency at which packets are delivered to their destination within their specified maximum latency. Packets that are dropped and do not reach their destination do not contribute to this metric.

D. Simulation Results

1) Random Networks: The performance metrics when different values of traffic load are generated at the source nodes are plotted in Figure 5. The curves and bars reflect the average...
values and the shaded areas (and error bars) represent the 95% confidence intervals.

Figure 5(a) shows that regardless of the traffic load, the ILP model is able to deliver all generated packets to destination, that is, the delivery ratio is 1 and the dropped packets are 0. This is achieved thanks to the fact that the model has high computational complexity and is able to access global network information both in terms of topology and traffic. Furthermore, as mentioned in Section III-B, only cases where the ILP model is able to find a feasible solution are considered in this analysis. On the other hand, the rest of the schemes have a delivery ratio that decreases as the traffic load increases. The reason for this is that the generated congestion starts to prevent the delivery of traffic to the destination. It should be noted that when a packet cannot be sent to its destination on time, all schemes delete that packet.

Figures 5(b) and 5(c) depict the packets that are discarded at each node under low traffic load (3 packets per node) and intermediate traffic load (6 packets per node), respectively. For a fixed traffic load, the original CGR-DelTime scheme shows the highest value of dropped packets. This packet loss occurs in each of the 10 nodes, although it is accentuated in nodes 6 to 10, which are those that generate traffic with maximum latency constraints (TTL = 20 s). With a lower packet loss, there are the CGR-MO schemes with weights 0.25, 0.5 and 0.75. The higher the weight (and importance) given to the number of hops, the better the congestion avoidance capability, and, therefore the lower the packet loss. Finally, CGR-Hops, which is equivalent to CGR-MO w = 1.0, is the scheme with the lowest packet loss and hence the highest delivery ratio.

Regarding the mean hops per packet metric (Figure 5(d)), CGR-Hops obtains the lowest values because it chooses routes with fewer hops. In this case, the ILP model results in a higher number of hops per packet because, in addition to satisfying the maximum latency constraint, it aims to deliver packets as quickly as possible, even if that means using more hops. The decreasing trend of hops per packet as the traffic load increases is because congestion forces the model to abandon early states and opt for later states with shorter routes in terms of hops.

In this context, energy efficiency is the measure of the number of successfully delivered packets per transmission. It is the inverse of the mean hops per packet metric and ranges between 0 and 1. Figure 5(e) shows that CGR-Hops is the most energy-efficient scheme, with the ILP model falling between CGR-DelTime and CGR-Hops. The model’s lack of efficiency can be attributed to its strategy of prioritizing quick delivery, which results in the use of many arcs in early states before moving to later states. In contrast, CGR-Hops opts for routes with fewer hops as long as they meet the maximum latency requirement. CGR-MO shows values between CGR-DelTime and CGR-Hops as the weight increases from 0.25 to 0.75, as expected.

Finally, as shown in Figure 5(f), the ILP model obtains the best metric in terms of delay per packet followed by CGR-DelTime, CGR-MO, and CGR-Hops. However, it is worth noting that this metric is averaging the delay for all generated packets regardless of their maximum latency requirement. If the mean delay is calculated separately for each type of traffic, we can expect the values to be lower for packets with lower values of TTL.

Figure 6 verifies this statement by separating the mean delay for packets with TTL = 20 s, from the mean delay for packets with TTL → ∞. Figure 6(a) shows that the curves for all schemes have close values for the entire range of traffic load and that these values are effectively between 0 and 20 seconds. In contrast, Figure 6(b) evidences that the curves have values more distant from each other. The decrease in mean delay when generating 10 packets per node happens because this metric only considers those packets that arrive at the destination, so those packets that are discarded due to congestion do not contribute to increase the metric.

In general, these results reveal that CGR-Hops delays traffic that can be delayed (i.e., traffic with higher TTL), thus alleviating congestion and allowing traffic that needs to be delivered earlier to reach its destination on time.

2) Walker-Delta Constellation: The results when the traffic load varies between 6 and 36 packets per source node are plotted in Figure 7. Since only one Walker-Delta constellation has been simulated, confidence intervals are not plotted. In
addition, since the ILP model requires extensive run times for scenarios of this size and time, its results are not included here.

The delivery ratio metric shows that even for low traffic load values, all routing schemes start with a packet loss between 15% and 20%. As expected, this loss increases with increasing traffic load and hence network congestion. Furthermore, it is observed that the greatest advantage of CGR-Hops over CGR-DelTime occurs at intermediate traffic load values. When the traffic load is very low, all routing schemes provide solutions

---

Fig. 7. Simulation results for Walker-Delta constellation.
that do not lead to competition for the same resources (contacts). When the traffic load is very high, the chances of providing good solutions decrease.

The mean hops per packet and energy efficiency metrics show behaviors similar to those obtained for the random networks scenario. CGR-Hops has the highest efficiency as it uses the least number of hops. The CGR-MO schemes obtain intermediate and increasing values of energy efficiency as the value of \( w \) increases. CGR-DelTime leads to the highest number of transmissions to deliver packets to the destination, thus obtaining low energy efficiency. However, a notable difference with the results obtained for random networks is that here the absolute values of the mean number of hops per packet are higher and the energy efficiency values are lower. This is explained by the fact that the simulated Walker-Delta constellation is sparser than the random networks with \( \delta = 0.2 \). Therefore, the routes that packets need to follow to reach their destinations are generally longer in terms of the number of hops.

Finally, the mean delay per packet metric reveals that CGR-Hops pays the cost of obtaining high energy efficiency by generating a higher mean delay per packet. However, to better appreciate which packets are delayed, the average delay per packet is plotted here again, distinguishing according to the TTL configured for each packet. Thus, it can be seen that packets with \( \text{TTL} = 3600 \text{ s} \) are delivered to their destination with similar times (and less than 3600 s), while the packets that are effectively more delayed are those that do not have maximum latency restriction (\( \text{TTL} \to \infty \)). A particular effect can be seen in Figures 7(d) and 7(e) when the traffic load increases to more than 30 packets per node. The curves of CGR-DelTime and CGR-MO, \( w = 0.25 \) appear to cross the curve of CGR-MO, \( w = 0.75 \). One possible explanation for this is that schemes such as CGR-DelTime, which strictly favor the delay metric, end up exacerbating congestion by trying to get all traffic to the destination sooner, it is also true that these schemes provide greater rerouting opportunities for traffic that is bottlenecked at intermediate nodes. On the other hand, it should be noted here again that since the mean delay per packet metric only considers those packets that actually arrive at their destination, it is not straightforward to compare between schemes when they provide different delivery ratio values.

IV. FUTURE DIRECTIONS

As future work, we envision further modifications to CGR to improve performance. In particular considering:

- **Load balancing:** When all nodes always use the shortest routes, it is likely that multiple routes will share more links, creating a bottleneck in terms of capacity and congestion. To address this issue, we propose different load balancing techniques by implementing round-robin on the \( K \) best routes.

- **Centrality:** In graph theory, the centrality metric measures how many routes pass through a link or node. To avoid congestion, we can use routes through links or nodes that do not have the highest centrality.

- **Machine learning:** We propose using machine learning models to estimate congestion on a link or node at a given time. We can then use these local estimates to carry out routing and avoid more congested routes but without under-utilizing them.

- **Uncertain contact plans:** Since contact plans cannot be perfectly implemented in practice, investigating the impact of contact plan uncertainty on traffic routing continues to be a research area that requires special attention.

V. CONCLUSION

In this work, we have proposed and evaluated adaptations to CGR to improve performance when sending traffic with different latency requirements. The results reveal that CGR currently has a limited capability to honor latency constraints. The reason is twofold: on the one hand, it only uses local traffic information. On the other hand, it strictly selects those routes with the lowest estimated delivery time without taking into account the number of hops involved. Since intermediate nodes also generate and carry traffic, this leads to congestion situations that prevent the effective fulfillment of latency requirements. To address these challenges, the proposed schemes explore different ways to consider both the estimated delivery time and the number of hops. The results show that the new schemes obtain higher delivery rates and better energy efficiency by using fewer hops per packet. In addition, the cost of these benefits is paid for by increasing the delay of those packets without latency constraints. Packets that need to arrive earlier continue to be delivered on time to their respective destinations. This opens the debate on different ways to improve performance without compromising the quality of service required by heterogeneous traffic.

REFERENCES

[1] M. Y. Abdelsadek et al., “Future space networks: Toward the next giant leap for humankind,” IEEE Trans. Commun., vol. 71, no. 2, pp. 949–1007, Feb. 2023.
[2] V. Cerf et al., “Delay-tolerant networking architecture,” IETF, RFC 4838, Apr. 2007.
[3] J. A. Fraire, O. De Jonckère, and S. C. Burleigh, “Routing in the space Internet: A contact graph routing tutorial,” J. Netw. Comput. Appl., vol. 174, Jan. 2021, Art. no. 102884.
[4] J. Segui, E. Jennings, and S. Burleigh, “Enhancing contact graph routing for delay tolerant space networking,” in Proc. IEEE Global Telecommun. Conf. (Globecom), 2011, pp. 1–6.
[5] E. Birrane, S. Burleigh, and N. Kasch, “Analysis of the contact graph routing algorithm: Bounding interplanetary paths,” Acta Astronautica, vol. 75, pp. 108–119, Jun./Jul. 2012.
[6] C. Caini, G. M. De Cola, and L. Persampieri, “Scheduling-aware bundle routing: Analysis and enhancements,” Int. J. Satellite Commun. Netw., vol. 39, no. 3, pp. 237–249, 2021.
[7] J. Edward, “Improving graph-based overlay routing in delay tolerant networks,” in Proc. IEEE Wireless Days, 2011, pp. 1–6.
[8] N. Bezirgiannidis, C. Caini, D. P. Montenero, M. Ruggieri, and V. Tsaoussidis, “Contact graph routing enhancements for delay tolerant space communications,” in Proc. 7th IEEE Adv. Satellite Multimedia Syst. Conf. 13th Signal Process. Space Commun. Workshop (ASMSSPSC), 2014, pp. 17–23.
[9] N. Bezirgiannidis, C. Caini, and V. Tsaoussidis, “Analysis of contact graph routing enhancements for DTN space communications,” Int. J. Satellite Commun. Netw., vol. 34, no. 5, pp. 695–709, 2016.
Peng Hu (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Queen’s University, Canada. He is currently a Research Officer with National Research Council Canada and an Adjunct Professor with the Cheriton School of Computer Science, University of Waterloo. His current research interests include satellite-terrestrial integrated networks, autonomous networking, and industrial Internet of Things systems. He has served as an Associate Editor of the IEEE CANADIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING, a Voting Member of the IEEE Sensors Council Standards Committee, and on the Organizing/Technical Committees of Industry Consortia and International Conferences/Workshops at IEEE GLOBECOM’23, IEEE ICC’23, IEEE GLOBECOM’21, IEEE PIMRC’17, and IEEE AINA’15.

Khaled Ahmed received the B.S. and M.Sc. degrees in wireless communications from Cairo University, Egypt, in 2010 and 2015, respectively, and the Ph.D. degree in optical communications and the Postdoctoral degree in machine learning for resource-limited devices from McMaster University, Canada, in 2019 and 2021, respectively. He is currently working as a Member Technical Staff II with the Systems Engineering Department, MDA, Sainte-Anne-de Bellevue, QC, Canada. His current research interests include RF and optical communications for terrestrial and satellite networks, optical beamforming networks, applied machine learning, and deep learning in networks.

Stéphane Martel received the Bachelor’s degree in electrical engineering from McGill University, Montreal, QC, Canada, in 2001. Until 2021, he worked with the Broadcast Television Industry, where he occupied multiple roles in software development and management. He specializes in network communications and embedded software development. In 2021, he joined MDA Satellite systems as a Research and Development Product Development Manager, where he contributes to the advancement of space communication technologies.

Guillaume Lamontagne received the B.Eng. and M.Eng. degrees from the École de Technologie Supérieure, Montreal, QC, Canada, in 2007 and 2009, respectively. His experience in satellite communications started through internships and research activities with the Canadian Space Agency in 2005, and the Centre national d’études spatiales, France, in 2006 and 2008. He joined MDA in 2009, and held various communication systems engineering and management positions before being appointed as the Director of Technology, Payloads in 2019. Through this role, he is leading MDA’s research and development activities for satellite communications as well as establishing the related long-term development strategy.