

ORIGINAL PAPER OPEN ACCESS

# A Load-Balancing Enhancement to Schedule-Aware Bundle Routing

Jason J. Kamps  | Filip Palunčić  | B. T. Maharaj 

Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa

**Correspondence:** Jason J. Kamps ([jason.kamps@tuks.co.za](mailto:jason.kamps@tuks.co.za))**Received:** 11 April 2024 | **Revised:** 29 November 2024 | **Accepted:** 6 January 2025**Funding:** This work was supported by the Sentech Chair in Broadband Wireless Multimedia Communications.**Keywords:** contact graph routing (CGR) | delay- and disruption-tolerant networking (DTN) | interplanetary internet | schedule-aware bundle routing (SABR)

## ABSTRACT

Delay- and disruption-tolerant networking (DTN) enables communication in networks afflicted by long propagation delays and sporadic connectivity. DTN routing techniques such as schedule-aware bundle routing (SABR) exist to route data bundles in deterministic networks, such as those found in deep-space environments, where node contacts are predictable. This article begins with an overview of DTN architecture and SABR. SABR's method of final route selection (forwarding rules) is closely examined. The article then addresses a limitation of SABR whereby the algorithm may overlook parallel channels, leading to network congestion. To mitigate this, an enhancement is proposed. This enhancement aims to optimize data bundle distribution across candidate routes in networks with parallel channels, thus alleviating congestion and enhancing overall network performance. This is achieved with simple modifications to SABR's forwarding rules to avoid the concentration of data bundles on a minority of node contacts. The enhancement is demonstrated through simulations in a reference scenario implemented in DtnSim.

## 1 | Introduction

Communication in deep space is challenging. Long propagation delays, intermittent connectivity, unidirectional links, and asymmetric bandwidth pose significant hurdles in establishing reliable communication. With increasing deep-space activity and the imminent emergence of the interplanetary Internet, developments have been made to support communication in the uniquely challenging environment of space.

Delay- and disruption-tolerant networking (DTN) [1] enables communication in networks lacking continuous connectivity. The Bundle Protocol [2, 3] is a prominent DTN protocol that transports units of information (bundles) from source to destination in a store-and-forward fashion. The Bundle Protocol is deployed as an overlay network, operating below the application layer.

The Bundle Protocol can operate in a variety of challenging networks, including terrestrial and space networks. Terrestrial DTNs are typically characterized by unpredictable links between nodes, such as with the movement of people and vehicles in a city. Space-based DTNs have predictable, scheduled links between nodes since the movement of satellites and celestial bodies can be calculated in advance. Routing in each of these network types is different. Bundle routing in terrestrial DTNs often relies on the replication and distribution of bundles to increase the likelihood of a bundle reaching its destination. Conversely, routing in space-based DTNs typically relies on the prudent forwarding of bundles from node to node with no duplication, exploiting known upcoming communication opportunities.

In space-based DTNs, contact graph routing (CGR) [4] is the most prominent routing algorithm and has received multiple revisions and enhancements. The CGR implementation in NASA's

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2025 The Author(s). *International Journal of Satellite Communications and Networking* published by John Wiley & Sons Ltd.

Interplanetary Overlay Network served as the baseline for the standardization of the algorithm at the Consultative Committee for Space Data Systems (CCSDS) under the name schedule-aware bundle routing (SABR) [5].

This article provides an overview of DTN architecture and the SABR specification as implemented in Unibo-CGR [6] (an independent implementation of SABR), followed by the presentation of an enhancement to SABR. This enhancement is designed to improve routing performance in networks that feature parallel channels. A simple reference scenario is introduced to demonstrate the enhancement with communication simulations performed using DtnSim [7], an event-driven DTN simulation tool.

## 2 | DTN Architecture

Propagation delays and intermittent connectivity of deep space make the use of traditional communication techniques impractical. Interplanetary distances impose a significant constraint on communication, as the propagation of electromagnetic waves, limited to the speed of light, leads to substantial delays. It can take minutes or hours for signals to reach locations within the Solar System. Besides making near-real-time communication impossible, large propagation delays cause traditional Internet protocols to fail or become exceptionally inefficient [8].

Planetary and spacecraft movements result in the frequent interruption of connections between communication nodes. For example, as satellites orbit, they are eclipsed by their planet, preventing communication with nodes on the opposite side of the planet (occultation) [9]. Since end-to-end connectivity is never guaranteed in DTN, messages must be transmitted iteratively from node to node until the messages reach their final destination. Messages are stored at nodes when an onward contact is not immediately available. A contact is a transmission opportunity between nodes. Contacts are unidirectional and ensure that any information transmitted during the specified time interval will be effectively received at the receiving node. The method of iterative transmissions is known as store-and-forward [10]. Intermediary DTN nodes must therefore provide storage capacity for messages before they are forwarded to the next node in the path. Furthermore, DTN allows automatic repeat requests [11] to originate from intermediate nodes, with no source-to-destination connection required for such retransmission requests.

Internet protocols are unlikely to work well in such extreme environments [12], and their possible configuration and enhancement to accommodate the challenges of deep space are to this date unresolved.

The Bundle Protocol is an implementation of DTN architecture, allowing data to be transmitted through networks that experience large propagation delays and intermittent connections between nodes. Data managed by the Bundle Protocol is organized into bundles: self-contained units of information that contain the actual data to be transmitted, along with additional metadata and control information to allow the bundle to make progress through the network. Bundles can be arbitrarily large.

The Bundle Protocol acts as an overlay network, operating above the transport layer in the Internet protocol stack. Through convergence layer adapters (CLAs), the Bundle Protocol is compatible with transport protocols such as TCP [13], UDP [14], as well as protocols that are better suited to DTNs, such as the Licklider Transmission Protocol [15]. The Bundle Protocol can therefore operate across heterogeneous networks, ensuring end-to-end connectivity even if each DTN hop employs a different CLA.

Version 6 of the Bundle Protocol enabled the transfer of custody of bundles from one node to another [2]. This is useful when the original sender becomes detached from the network. The node with custody of a particular bundle becomes responsible for re-transmissions until custody is transferred. It should be noted that the custodial retransmission mechanism was removed in version 7 of the Bundle Protocol, but it may return, with modifications, as an extension block.

Bundles consist of various blocks of data, analogous to packet headers and payloads. The primary block contains information such as the destination end-point identifier which is needed for bundle routing. Bundles can be formatted with extension blocks such as CGR-EB [16] which encodes end-to-end message paths to enable graph-based overlay routing for a variety of networks.

Bundles may be fragmented into smaller bundles during transmission [12]. Each fragment bundle inherits the original bundle's primary block. Fragments can be further fragmented or reassembled, allowing adaptability to the available volume of contacts.

Routing in DTN is not trivial. Contact information must be made known to or must be discovered by DTN nodes to inform routing decisions. It may be impossible to know the state of the entire network in real time due to propagation delays. In DTN routing, the most desirable routing objective may not be immediately obvious [17]; however, increasing the probability of bundle delivery and reducing delivery delay are important.

## 3 | Unibo-CGR SABR Algorithm

Given the challenges of DTN, specialized routing techniques are required. In deterministic networks, CGR is a prominent algorithm used for the routing of Bundle Protocol bundles. CGR operates using contact plans which characterize the time-varying connectivity in a network. A contact plan is a list of predicted (unidirectional) contacts, associated with delays and data rate information. In CGR, each node in the network constructs a contact graph from contact plan messages, allowing each node to compute logical bundle forwarding decisions automatically. Despite its suitability in deterministic networks, CGR has some known limitations. CGR has been found to overlook parallel channels, which if used could improve path redundancy and increase bandwidth [18].

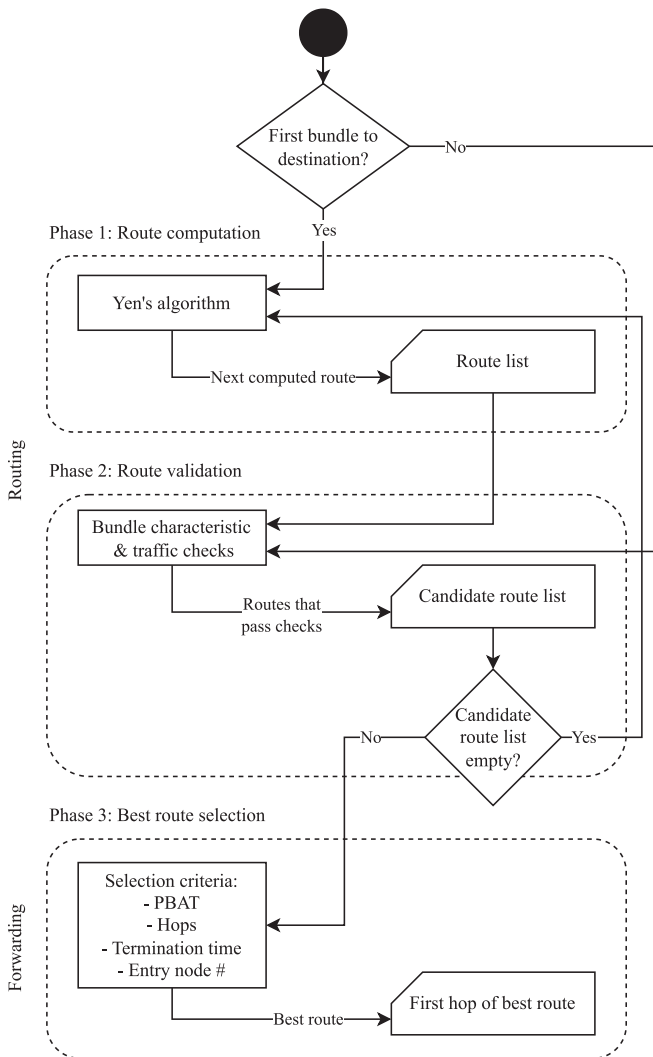
CGR is especially advantageous for networks in space, as it can take advantage of the highly predictable contacts between nodes [19]. An overview of a CGR implementation is presented here. More specifically, an implementation of the latest version of CGR standardized by the CCSDS is explored. This version is known as SABR.

SABR, as implemented in Unibo-CGR [6] and summarized in Figure 1, has three main stages:

1. **Planning:** A contact graph is constructed from a contact plan to represent the evolving communication opportunities within the network.
2. **Routing:** Using the contact graph, routes that offer the earliest arrival time are computed and placed in a candidate route list after being validated.
3. **Forwarding:** The best path from the candidate route list is selected for each bundle based on current conditions. Bundles with the same destination may have different routes depending on the time they are transmitted, their priority, and local traffic conditions.

### 3.1 | Planning

In certain networks, especially space-based networks, the upcoming communications opportunities between nodes are known. These periods are contacts: intervals during which data can be transmitted from one node to another at a certain rate.



**FIGURE 1** | Overview of SABR, as implemented in Unibo-CGR [6].

The entries in a contact plan take the form:

```
contact <start time> <end time> <source> <destination> <rate>
```

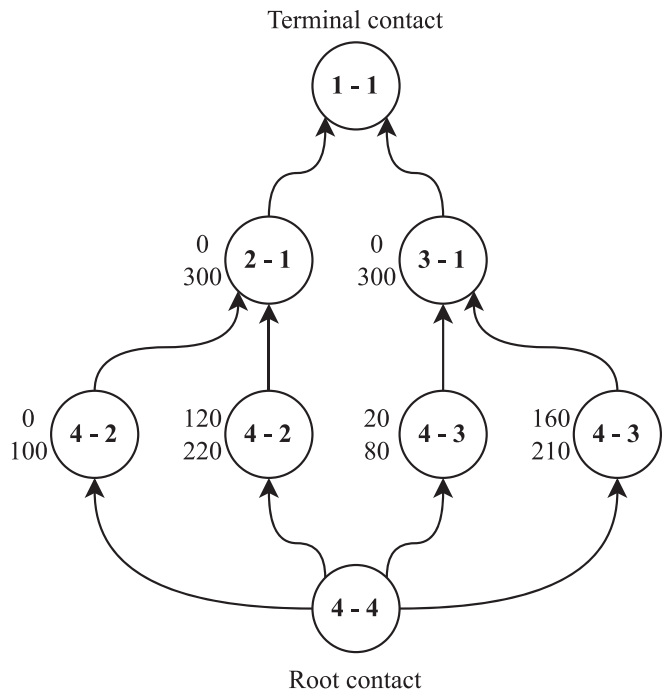
Contact plans are computed in advance and subsequently distributed to all nodes. This computation is typically performed at a central location such as mission control. A contact graph is a direct acyclic graph that efficiently represents the time-evolving topology of a network, can be constructed from a contact plan. An example contact graph, illustrated in Figure 2, represents the contact plan given in Table 1 for bundles starting at Node 4 with a destination of Node 1. Notice that the vertices of the contact graph are contacts, not network nodes. The edges show periods where data may need to be stored between contacts.

### 3.2 | Routing

#### 3.2.1 | Phase 1: Route Computation

In CGR, a route is a sequence of contacts between nodes that enable the transfer of data from the source to the destination. It is not typically sufficient to compute a single route to each destination. Multiple routes are required since routes expire, have limited volume, and communication links may fail unexpectedly. Contacts have a finite duration, thus the validity of any single route ends with the expiration of any one of its contacts. A contact may be exhausted for bundles of a certain priority, and an alternative must be used. Finally, multiple routes allow for redundancy and, if exploited correctly, additional concurrent bandwidth.

The contact graph is searched to find the shortest paths between source and destination. The details of this search are not defined in the SABR standard and are left as an



**FIGURE 2** | A contact graph representation of the contact plan given in Table 1, for routing to Node 1 from Node 4.

**TABLE 1** | Reference scenario contact plan.

Start time (s)	End time (s)	Source node	Destination node	Data rate (kb/s)
0	100	5	2	256
0	100	4	2	256
20	80	5	3	256
20	80	4	3	256
120	220	5	2	256
120	220	4	2	256
160	210	5	3	256
160	210	4	3	256
0	300	2	1	256
0	300	3	1	256

implementation matter. Several algorithms exist to compute the route list, including Yen's  $K$  Shortest Path algorithm which is mentioned in the standard and is common in modern CGR implementations.

Yen's algorithm [20] returns the  $K$  shortest paths between nodes in a graph. For  $K = 1$ , any shortest path algorithm is used, such as Dijkstra's algorithm [21]. To find additional short paths ( $K > 1$ ), routes are systematically explored by selectively removing edges one at a time to force alternative paths.

While Yen's algorithm is used in this investigation, alternatives include anchoring, first-ending, first-depleted, and SPSN. Like Yen's algorithm, anchoring, first-ending, and first-depleted all employ a series of Dijkstra searches to compute multiple routes. The contact graph is modified between successive searches to induce the computation of a new route.

After the first route has been computed, an approach to compute alternative routes is to suppress the first contact of the initially computed route. This approach, however, can lead to many undiscovered routes if the initial contact has a long duration. The anchoring approach, where a long contact is protected from suppression, allows CGR to find routes that would otherwise be hidden [22].

The first-ending approach suppresses the contact with the earliest termination time of the last-found route while the first-depleted approach suppresses the contact whose available capacity would become fully exhausted first [23]. This method emulates the saturation of routes, removing contacts as they become fully booked.

The Shortest-Path Tree Approach for Routing in Space Networks (SPSN) is a computationally efficient alternative to CGR shown to scale better to large networks [24]. SPSN determines routes using a spanning-tree approach in lieu of a contact graph approach. Efficiency is gained by using deterministic

capacity-oriented search which has a lower computational complexity relative to Yen's algorithm.

At first, only a single route is computed and stored in a list of possible routes. Yen's algorithm, for example, can then be used to discover additional routes when such routes are required.

Note that the routes found by SABR are an ordered list of contacts. This is in contrast to traditional Internet routing where routes consist of a list of nodes.

### 3.2.2 | Phase 2: Route Validation

Next, each route in the route list is examined to determine if it is suitable given the characteristics of the current bundle and local traffic information. A route is unsuitable if it meets any of the following criteria:

- The best-case delivery time is after the bundle's expiration time.
- The route's first node is in the list of excluded nodes (avoid backward propagation).
- The route includes the current node at a later point in time (avoid routing loops).
- The route's existing backlog results in an earliest transmission opportunity that is after the end of the route's initial contact.
- The route's projected bundle arrival time (PBAT) is after the bundle's expiration time.
- A contact in the route is expected to have insufficient capacity for the bundle given its priority (the route is depleted).

All suitable routes are placed in the candidate route list. At this stage, if the candidate route list is empty, Yen's algorithm is performed again to find alternative routes.

### 3.3 | Forwarding

After establishing a list of candidate routes, the best route is chosen according to the following criteria:

- The route with the earliest PBAT.
- If equal, the route with the fewest number of contacts (fewest hops).
- If equal, the route with the latest termination time.
- If equal, the route with the smallest entry node ID (arbitrary selection).

The route with the earliest PBAT is thus selected, and if there are multiple routes with the same PBAT, tiebreak rules are employed to select a single route.

Bundles marked critical are forwarded to all neighboring nodes for which there is a candidate route to the destination. This

process requires multiple executions of route computation and route validation to find additional candidate routes.

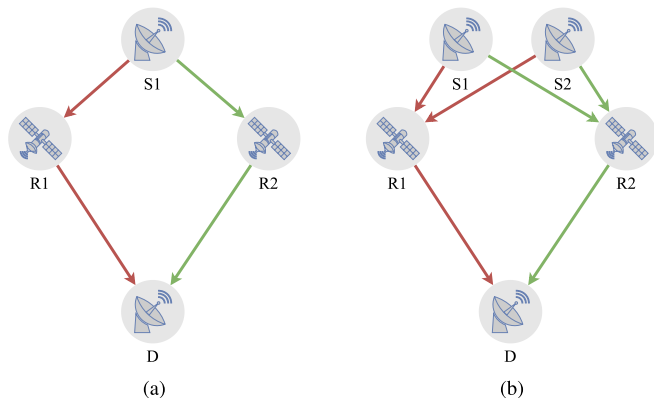
#### 4 | SABR Limitation

SABR is known to have a limitation that prevents the exploitation of latent bandwidth available in networks that contain parallel channels [18]. In these networks, SABR implementations tend to select a single particular route for bundles despite the availability of other similar routes. This is due to the fact that the candidate route list for a destination is only populated on demand (by going back to phase 1) when there are no candidate routes for the current bundle. Until there is at least one candidate route, alternative routes that are potentially better are not explored. This may cause downstream congestion. Space network topologies may be particularly vulnerable to this limitation as many proposed interplanetary networks consist of a limited number of long-haul links between planetary systems, with denser networks on each side.

Consider the following simple scenario as an example: Two Mars bases on opposite hemispheres of Mars have access to two relay satellites in areostationary orbits, illustrated in Figure 3b. These relays communicate with a ground station on Earth. In this example, there is no direct communication channel between the Mars bases. Similarly, there is no direct communication channel between the relays.

Consider the network in Figure 3a which contains a single source node. The propagation delay of both links between the source node and relays is equal. S1 has two candidate routes to D: red and green. There is no significant difference between the red and green routes. Under SABR, the red route would be chosen for all bundles as this route would be the only computed route. In Unibo-CGR, if there are multiple candidate routes, the red route would be chosen for all bundles as it has the smallest entry node number (see forwarding rules in Section 3.3). Multiple candidate routes can be induced with the one-router-neighbor enhancement described in [6].

Figure 3b shows the network with the addition of a second source node, S2. Routing decisions at S2 are identical to those at S1. Thus, all bundles originating at S2 are routed along the red route. With all bundles from both S1 and S2 directed along



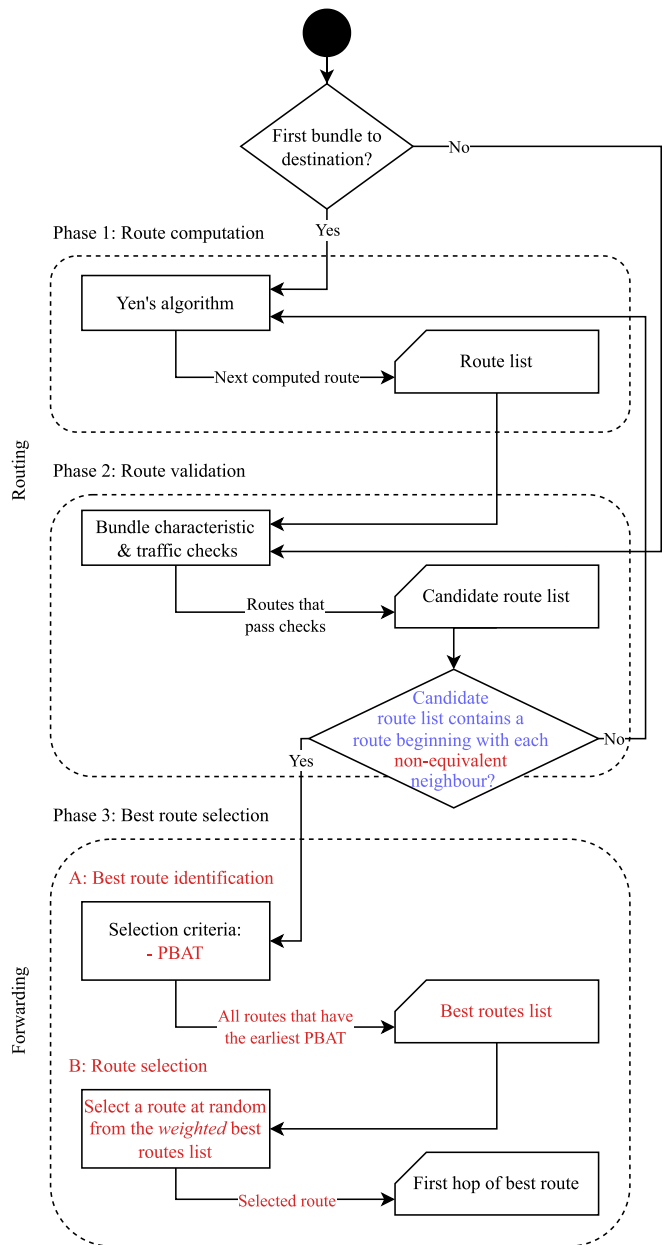
**FIGURE 3** | Simple networks comprising source node(s) (S1 and S2) with communication routes to a destination (D) via relays (R1 and R2).

the red routes, the long haul link between R1 and D has the potential to become congested while the equivalent green link between R2 and D remains unused.

Modifications to Unibo-CGR are proposed to uniformly route bundles when multiple similar routes are available. These modifications aim to distribute bundles among all available routes thereby reducing congestion and improving network resource utilization.

#### 5 | Load-Balancing Enhancement

The proposed load-balancing enhancement is a modification<sup>1</sup> to Unibo-CGR [6], an independent implementation of SABR. The enhancement is summarized in Figure 4 and can be



**FIGURE 4** | Overview of SABR, as implemented in Unibo-CGR [6], with the load-balancing enhancement. Blue text indicates enhancements available in Unibo-CGR. Red text indicates the proposed load-balancing enhancement.

compared to the standard operation of Unibo-CGR given in Figure 1.

SABR selects the same route for bundles with the same destination, provided that the route has sufficient capacity. This becomes particularly problematic if there are multiple data-producing source nodes since each source may independently select routes that use the same contacts, resulting in congestion at these contacts. In DTN, there is no guarantee that nodes are aware of up-to-date traffic information throughout the entire network, and it is often impossible to know in real time due to long propagation delays. Thus, improved route selection is required to reduce potential congestion.

The load-balancing enhancement aims to distribute bundles between all similar candidate routes. This is achieved by dividing the forwarding stage into two parts: (A) best route identification and (B) route selection.

## 5.1 | Forwarding

### 5.1.1 | Part A: Best Route Identification

In ordinary SABR, the best route of all candidate routes is the route with the earliest PBAT. When there are multiple routes with the same PBAT, the tiebreak rules result in the same route being selected for all bundles. With load balancing, all routes with similar PBATs are considered, and instead of tiebreak rules, a weighted round-robin [25] style distribution technique is used.

The route with the earliest PBAT is identified from the candidate route list. All other candidate routes are subsequently evaluated and routes with similar PBATs are placed in a list of best routes. An adjustable tolerance is used to determine the similarity threshold. A tolerance of 5%, for example, would allow routes with a PBAT that is no later than the best route's PBAT + 5% of the best route's duration.

### 5.1.2 | Part B: Route Selection

To avoid selecting the same route during each execution of the algorithm, the tiebreak rules are replaced by a random selector based on a random number generator. This results in the even distribution of bundles between all similar routes, reducing the likelihood of overloading a single contact. If each node distributes its outgoing bundles between all best routes, traffic will be spread more evenly, making efficient use of all network resources and avoiding the concentration of bundles at a minority of contacts.

The distribution of bundles can be improved by assigning weights to each route. Routes with more capacity should be chosen more often to avoid saturating lower-capacity routes. As such, the best routes are weighted according to their route volume limit. This metric represents the expected available capacity of the most constrained contact (the bottleneck contact) of the entire route, calculated using the local traffic information at the current node.

It is important to consider that introducing randomness into an otherwise deterministic algorithm may not always be ideal. Alternatively, the same outcome could be achieved by selecting the least-congested route (based on the percentage of remaining volume) for each bundle. The effectiveness of this deterministic arbitration method should be validated in future research.

## 5.2 | Routing

Another modification is made in the routing stage, to improve the routes available for selection in the forwarding stage. During route computation, the route list is appended not with only the next fastest route to the destination but with the next fastest route that begins with each neighbor of the current node. This is achieved by enabling the one-route-per-neighbor enhancement available in Unibo-CGR. This is intended to increase the quantity of candidate routes available for selection.

In SABR, the route computation stage (Yen's algorithm) is not re-entered unless necessary. In many implementations, only one suitable route exists in the route list, with additional routes computed only if there is not a single suitable route in the route list. Route computation stops as soon as a newly computed route becomes suitable. The one-route-per-neighbor enhancement aims to produce a multiplicity of candidate routes by forcing the computation of a candidate route for each neighbor. The route computation stage is re-entered with each neighbor as the source. This improves the chances of selecting a better route in the forwarding stage.

The routing step also excludes routes that begin with equivalent neighbors. An equivalent neighbor is a neighboring node that can be reached with negligible delay and whose onward contacts are similar to the current node. This typically occurs when multiple nodes occupy a similar location. In the following reference scenario, Relays 1 and 2 would be considered equivalent if they shared an instantaneous link. This restriction is implemented to prevent repeated transmissions of bundles between equivalent nodes in instances where several nodes are in close proximity to each other.

## 5.3 | Other Techniques

Techniques have been developed for CGR to avoid congestion and manage volume. Such techniques include centralized contact plan design, which utilizes a priori knowledge of network topology and expected traffic to create customized contact plans for each node. This approach aims to reserve network capacities and avoid congestion when nodes later apply routing algorithms [26].

Local Path-Aware Contact Graph Routing (LPA-CGR) considers the capacity of each local contact when forwarding packets, avoiding congestion on local routes. However, it may struggle with nonlocal traffic. To address this, Global Path-Aware Contact Graph Routing (GPA-CGR) takes into account the global network state, designing contact plans that account for predicted traffic across the entire network [26]. This method involves

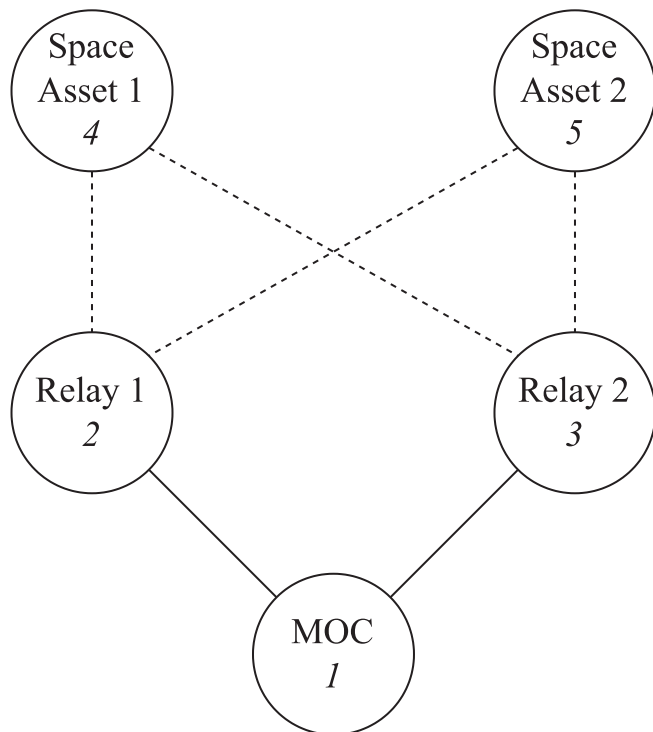
centralized computation of route tables and flow assignments. Congestion-free routing strategies, like GPA-CGR, create routes by considering all potential traffic flows and pre-allocating capacities to ensure no single channel is overburdened.

The load-balancing enhancement to SABR differs from centralized route computation approaches by computing routes locally at each node. This can be beneficial in networks with unpredictable traffic generation behavior and long propagation delays that hinder real-time knowledge of the complete network. Similar to LPA-CGR, the load-balancing enhancement considers the available capacity of each route but differs by then distributing bundles in a weighted round-robin fashion between the routes according to the available capacities.

## 6 | Reference Scenario

A simple reference scenario is used to highlight the effect of load balancing on networks that exhibit symmetry. Although this reference scenario does not represent any particular existing/future scenario, it has some resemblance to the Martian scenarios of [27] and [28] and is inspired by the reference scenario of [6]. The symmetry in this reference scenario is useful to evaluate SABR's route selection when presented with parallel channels.

Consider a scenario where two space assets are streaming data back to a mission operations center (MOC) on Earth. Two relays exist to bridge the gap between the spacecraft and Earth. This simple scenario is depicted by the topology in Figure 5.



**FIGURE 5** | Reference scenario topology. Dotted lines indicate intermittent links. Solid lines indicate continuous links.

The reference scenario's contact plan is given in Table 1 and visualized in Figure 6.

Note that

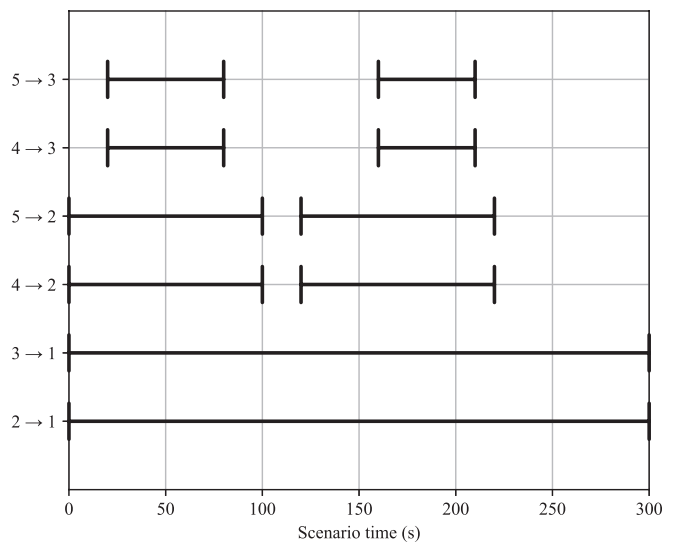
- The continuous contacts are declared as long contacts that span the duration of the scenario.
- The contacts are unidirectional and only downlink (space assets to MOC) contacts are represented in the table.
- The propagation delay of all contacts is set at 5 s, for simplicity.
- The contacts between both relays and the MOC have the same start and end times.

## 7 | Simulations

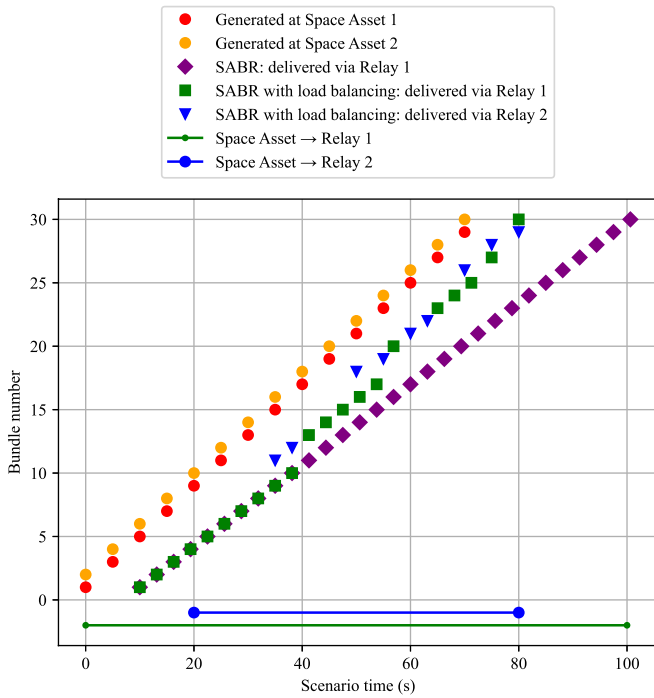
The reference scenario is configured in DtnSim [7] to perform the requisite networking simulations. At each space asset, 15 bundles of 100 KB are produced over the first 75 s of the scenario, for an effective data generation rate of 320 kb/s. The downlink progression of each bundle is logged for analysis.

First, a baseline is established by executing the scenario without the load-balancing enhancement. The results in Figure 7 reveal that all bundles from both space assets are routed via relay 1.

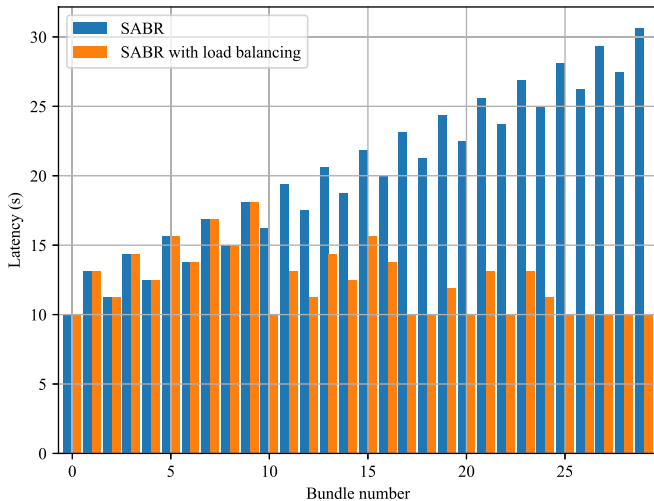
Without any queueing delays impacting PBAT, each source node selects Relay 1 as the next node for each bundle, despite the availability of the nested contact to Relay 2. The tiebreak rules result in all bundles being routed via Relay 1 as its contact has a later termination time. Notice that bundle latencies increase over time due to congestion at Relay 1. Next, the scenario is executed with load balancing enabled. A tolerance of 0% is set to only accept candidate routes with identical PBATs to the best route. In this ideal scenario, all routes from space assets to the MOC have identical PBATs. Figure 7 shows that the bundles are now routed via both relays.



**FIGURE 6** | Visualization of the reference scenario contact plan.



**FIGURE 7** | Generation and delivery times series. Bundles routed by SABR are only delivered via Relay 1. Bundles routed by SABR with the load-balancing enhancement are routed via both Relays 1 and 2.



**FIGURE 8** | Delivery latency of bundles routed by SABR in comparison to bundles routed by SABR with the load-balancing enhancement.

With the load-balancing enhancement, bundles are distributed between the two relays, reducing congestion at relay 1, and reducing the average bundle delay. Notice that all bundles are delivered within the first 80 seconds of the simulation, compared to 100 s without load balancing. The average bundle delivery time in the reference scenario with load balancing is 24.7 s, compared to 40.6 s with traffic spreading. This reveals an average delivery time gain of 15.9 s per bundle. The enhancement's effect on delivery latency is shown in Figure 8. Notice that after the introduction of Relay 2, the delivery latencies of bundles routed by SABR with the load-balancing enhancement are lower than that of standard SABR.

When the nested contact to Relay 2 begins, bundles now have two concurrent routes to the MOC. The weighted random distribution of bundles between these routes results in a more efficient use of the overall network. Load balancing causes seven bundles in the reference scenario to be routed via Relay 2 instead of Relay 1, reducing congestion on the Relay 1 route. In this simple scenario, a very small number of bundles are produced, resulting in a seemingly uneven distribution between the routes: only 7 of a possible 22 bundles are routed via Relay 2. In practice, when many bundles are produced, the law of large numbers ensures that the quantity of bundles assigned to each route is proportional to the route's capacity.

The use of the same route is preferred for bundles belonging to a media flow, to prevent disordered delivery. The load-balancing enhancement, however, can cause disordered delivery as seen in Figure 7. While this is not desirable for bundles belonging to a media flow, it remains compliant with Bundle Protocol RFCs [6, 29].

## 8 | Conclusion

Communication in deep space presents unique challenges including long propagation delays and intermittent connectivity. The Bundle Protocol and DTN routing specifications such as SABR have been developed to address these challenges. An overview of SABR is presented, along with a limitation of the specification: SABR may overlook parallel channels, leading to potential congestion and inefficient resource utilization in networks with similar routes.

A load-balancing enhancement to SABR is proposed to distribute bundles between all available similar routes. Simulations of a reference scenario demonstrated that the enhancement effectively distributed bundles between routes, reducing congestion and improving average bundle latency.

The proposed enhancement offers a promising improvement to routing decisions of SABR implementations in DTNs, ensuring more efficient utilization of network resources and improved communication performance in deep space environments. An evaluation of the enhancement's computational overhead is identified as an avenue for further investigation. Future work could also include the evaluation of the performance of the load-balancing enhancement in larger networks with significantly more nodes.

### Acknowledgments

This work was supported by the Sentech Chair in Broadband Wireless Multimedia Communications.

### Conflicts of Interest

The authors declare no conflicts of interest.

### Endnotes

<sup>1</sup>The modified code can be found here: <https://bitbucket.org/dtn-sabr-load-balancing/dtnsim>.

## References

1. S. Burleigh, A. Hooke, L. Torgerson, et al., "Delay-Tolerant Networking: An Approach to Interplanetary Internet," *IEEE Communications Magazine* 41, no. 6 (2003): 128–136.
2. K. Scott and S. Burleigh, "Bundle Protocol Specification," Request for Comments, (RFC Editor, 2007), <https://www.rfc-editor.org/info/rfc5050>.
3. S. Burleigh, K. Fall, and E. J. Birrane, "Bundle Protocol Version 7," Request for Comments, (RFC Editor, 2022), <https://www.rfc-editor.org/info/rfc9171>.
4. S. Burleigh, "Contact Graph Routing," (2010), Internet Engineering Task Force. Internet-Draft draft-burleigh-dtnrg-cgr-01, <https://datatracker.ietf.org/doc/draft-burleigh-dtnrg-cgr>.
5. Consultative Committee for Space Data Systems (CCSDS), "Schedule-Aware Bundle Routing, Recommended Standard," (2019), Recommended Standard, Issue 1, <https://public.ccsds.org/Pubs/734x3b1.pdf>, CCSDS 734.3-B-1, Blue Book.
6. C. Caini, G. M. De Cola, and L. Persampieri, "Schedule-Aware Bundle Routing: Analysis and Enhancements," *International Journal of Satellite Communications and Networking* 39, no. 3 (2021): 237–249.
7. J. A. Fraire, P. Madoery, F. Raverta, J. M. Finochietto, and R. Velazco, "DtnSim: Bridging the Gap Between Simulation and Implementation of Space-Terrestrial DTNs," in *2017 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, (Henaes, Madrid, Spain, 2017): 120–123.
8. S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, "When TCP Breaks: Delay- and Disruption-Tolerant Networking," *IEEE Internet Computing* 10, no. 4 (2006): 72–78.
9. D. Morabito and R. Hastrup, "Communicating With Mars During Periods of Solar Conjunction," in *IEEE Aerospace Conference Proceedings*, Vol. 3, (Big Sky, MT, USA, 2002): 3.
10. C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, "Delay- and Disruption-Tolerant Networking (DTN): An Alternative Solution for Future Satellite Networking Applications," *Proceedings of the IEEE* 99, no. 11 (2011): 1980–1997.
11. Shu Lin, D. J. Costello, and M. J. Miller, "Automatic-Repeat-Request Error-Control Schemes," *IEEE Communications Magazine* 22, no. 12 (1984): 5–17.
12. A. McMahon and S. Farrell, "Delay- and Disruption-Tolerant Networking," *IEEE Internet Computing* 13, no. 6 (2009): 82–87.
13. M. Demmer, J. Ott, and S. Perreault, "Delay-Tolerant Networking TCP Convergence-Layer Protocol," RFC 7242. Internet Engineering Task Force, (2014).
14. H. Kruse and S. Ostermann, "UDP Convergence Layers for the DTN Bundle and LTP Protocols," draft-irtf-dtnrg-udp-clayer-00. Internet Engineering Task Force, (2008).
15. S. Burleigh, S. Farrell, and M. Ramadas, "Licklider Transmission Protocol - Specification," (RFC Editor, 2008), <https://www.rfc-editor.org/info/rfc5326>.
16. E. J. Birrane, "Improving Graph-Based Overlay Routing in Delay Tolerant Networks," in *2011 IFIP Wireless Days (WD)*, (2011): 1–6.
17. S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, (Portland, OR, USA, 2004): 145–158.
18. R. Lent, "Evaluation of Cognitive Routing for the Interplanetary Internet," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, (Taipei, Taiwan, 2020): 1–6.
19. J. A. Fraire, O. De Jonckère, and S. Burleigh, "Routing in the Space Internet: A Contact Graph Routing Tutorial," *Journal of Network and Computer Applications* 174 (2021): 102884.
20. J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Science* 17, no. 11 (1971): 712–716.
21. E. W. Dijkstra, "A Note on Two Problems in Connexion With Graphs," *Numerische Mathematik* 1, no. 1 (1959): 269–271.
22. J. A. Fraire, P. Madoery, S. Burleigh, et al., "Assessing Contact Graph Routing Performance and Reliability in Distributed Satellite Constellations," *Journal of Computer Networks and Communications* 2017 (2017): 1–18.
23. J. A. Fraire, P. G. Madoery, A. Charif, and J. M. Finochietto, "On Route Table Computation Strategies in Delay-Tolerant Satellite Networks," *Ad Hoc Networks* 80 (2018): 31–40.
24. O. De Jonckère and J. A. Fraire, "A Shortest-Path Tree Approach for Routing in Space Networks," *China Communications* 17, no. 7 (2020): 52–66.
25. J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM* 19, no. 2 (1972): 248–264.
26. P. G. Madoery, J. A. Fraire, and J. M. Finochietto, "Congestion Management Techniques for Disruption-Tolerant satellite networks," *International Journal of Satellite Communications and Networking* 36, no. 2 (2018): 165–178.
27. G. K. Noreen et al., Integrated Network Architecture for Sustained Human and Robotic Exploration, in *IEEE Aerospace Conference Proceedings*, (Big Sky, MT, USA, 2005): 1266–1285.
28. N. Alessi, C. Caini, T. de Cola, S. Martin, and J. P. Mayer, "DTN Performance Analysis of Multi-Asset Mars-Earth Communications," *International Journal of Satellite Communications and Networking* 40, no. 1 (2022): 11–26.
29. N. Bezirgiannidis, C. Caini, and V. Tsaoussidis, "Analysis of Contact Graph Routing Enhancements for DTN Space Communications," *International Journal of Satellite Communications and Networking* 34, no. 5 (2016): 695–709.