

# Multi-Objective Cognitive Routing in Space DTNs

Ricardo Lent  
University of Houston  
Houston, Texas, USA  
Email: rlent@uh.edu

**Abstract**—In space networking and other challenging environments, optimizing bundle routing is crucial. Traditional routing objectives focus on finding the shortest path or achieving the earliest delivery time. However, with the growth of these networks in size and diversity of applications, there is a need to expand the scope of routing objectives. In this paper, we explore a decentralized bundle routing method with multiple objectives. These objectives include bundle response time, loss ratio, and a general cost that reflects the monetary cost rate associated with the utilization of network links. We also consider that certain links may be offered as on-demand or reserved services. To address these challenges, we propose a combined routing objective and study its performance. We then apply this objective to determine learning rewards for a reinforcement learning agent. Our results, based on a proof-of-concept using a standard Delay-Tolerant Networking (DTN) implementation, demonstrate the effectiveness of the proposed method. By accommodating multiple objectives, the method offers a flexible and adaptive solution for online routing optimization.

**Index Terms**—Delay-Tolerant Networking, Space Communications, Routing Optimization, Cognitive Networking, Reinforcement Learning

## I. INTRODUCTION

The performance of communication networks in space and other challenging environments is constrained by various factors. Among these, significant propagation delays and dynamic edge conditions arise from relative node mobility, solar activity, atmospheric conditions, and scheduling decisions made by mission control. These factors induce considerable variability in the channel performance over time, and certain links may be rendered unavailable for extended periods. Routing in this delay-tolerant networking (DTN) context is a difficult task but widely addressed by Contact Graph Routing (CGR) and provided as Schedule-Aware Bundle Routing by the Consultative Committee for Space Data Systems (CCSDS, 734.3-B-1). Given that in most cases it is possible to predict the node proximity (e.g., from orbit calculations or ephemeris), the knowledge of the future contact opportunities allows defining a contact graph where the nodes represent the possible contacts with edges linking the feasible contacts having the same end and start node respectively. The link feasibility depends on the timing of the contacts. A graph traversal yields the next-hop for a bundle in this approach.

CGR finds paths with the nearest delivery time, but its effectiveness is curbed by network congestion, which is not fully included in the computation. As such, the approach is mainly suitable for light network traffic. To deal with the problem, extensions to the basic CGR have been suggested,

such as CGR-ETO [1], which adds local queueing delays to the transmission time estimations. However, no routing customization exists that could prevent bundles from going through high-loss links. Losses that could not be recovered from or avoided by the lower layers are mitigated by the custody transfer mechanism of the Bundle Protocol (BP). As the network evolves and the commercialization of outer space continues, price-related metrics become relevant, i.e., how to customize routing to achieve the best out of the net monetary cost. In addition to whether metrics other than the delivery time should be considered when routing bundles, the question is how to formulate multi-criteria routing objectives that can achieve the optimal trade among multiple metrics. This study offers two main contributions:

- 1) Formulation and analysis of a multi-objective routing cost that includes the bundle response time, loss ratio, and other costs expressed as the monetary cost rate of links, as needed. We examine the Pareto fronts and the relevance of the proposed formulation to identify suitable trade-offs among the metrics of interest.
- 2) We applied the proposed formulation to cognitive DTN routing by suggesting and implementing extensions to the Cognitive Space Gateway (CSG) [2]. The CSG applies reinforcement learning and spiking neural networks (SNN) to the bundle routing optimization based on the minimum response time, and our proposed formulation enables the achievement of multi-criteria objectives. Experimental results using an implementation of the CSG for NASA's HDTN [3] demonstrate that the method can effectively identify a suitable trade-off among the selected metrics.

## II. RELATED WORKS

Close research to the current work is related to Schedule-Aware Bundle Routing (SABR), which is rooted in CGR. CGR has been designed to minimize bundle delivery times [1]. However, the lack of congestion information renders the outcome similar to hop-count routing. Variations of this idea have been proposed involving the partial inclusion of local queueing delays [4] and the allowance of opportunistic contacts modeled via a confidence metric [5]. Energy capacity concerns for nanosatellites have also been raised [6] along with scalability concerns [7] and the possibility of routing for multigraphs [8]. While CGR is designed for decentralized operation, several studies indicate a potential performance gap compared to a centralized approach (e.g., see [9], [10]), especially in scenarios allowing for immediate forwarding [11]

or utilizing mixed-integer linear programming [12]. This work stands apart from prior research through its application of machine learning and multi-objective routing.

### III. SYSTEM MODEL AND ROUTING OBJECTIVES

A delay-tolerant network can be represented as a dynamic graph  $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$ , where both the set of nodes  $\mathcal{V}(t)$  and the set of edges  $\mathcal{E}(t)$  are time-dependent. This dynamic graph is utilized to facilitate the transmission of one or more bundle flows, which may experience delays due to intermittent connectivity and network disruptions.

We are addressing the problem of a routing agent  $i$  that autonomously selects the next hop  $j$  for a bundle (or set of bundles) to reach a destination  $d$  with a minimum cost that depends on time-varying characteristics of the path  $P = (s, \dots, i, j, \dots, d)$  that the bundle follows. Here,  $s$  represents the source of the path. The cost of the path is subject to fluctuations due to the dynamic network, which the routing agent must take into account while making routing decisions.

The routing agent  $i$  does not have control over or the ability to modify the preceding path  $(s, \dots, i)$ , but it can influence the selection of the subsequent path to the destination  $d$  by choosing the next node  $j$  from the available set of current or future contacts with other nodes, denoted as  $j \in 1, \dots, J$ . The remaining path cost  $\phi_j$  is of particular relevance from the perspective of the routing agent  $i$ .

When considering the macroscopic behavior of a bundle flow traveling towards the destination  $d$  and passing through node  $i$ , the routing decisions made by the agent result in the flow being divided into  $J$  parts. Specifically, let  $\lambda_j$  denote the flow fraction sent to node  $j$ , and  $\lambda_T$  represent the total flow. Then, we have  $\lambda_T = \sum_j \lambda_j$  and  $p_j = \lambda_j / \lambda_T$  is the fraction of traffic directed towards node  $j$ . It is important to note that a routing agent may choose to send a bundle to a node that is not currently reachable. In this case, the bundle is buffered until transmission, contingent on available storage capacity. The objective is to minimize the average cost from  $i$  by defining a policy  $\mathbf{P}$ , where  $p_j \in \mathbf{P}$  is the fraction for traffic sent to  $j$  with a class of service (CoS)  $q$ :

$$\Phi_i^{(q)} = \min_{\mathbf{P}} \sum_{j; (i,j) \in \mathcal{E}} p_j \Phi_j^{(q)} (\lambda_j + F_j) \quad (1)$$

Here,  $\Phi_j^{(q)}$  is the cost function associated with sending the desired bundle's class-of-service  $q$  to the destination via node (or link)  $j$  with  $\mathcal{E}$  denoting the union of  $\mathcal{E}(t)$  for all  $t$ . Each cost  $\Phi_j^{(q)}$  depends on the traffic to be applied  $\lambda_j$  and the total traffic  $F_j$  produced by other flows.

#### A. Multi-objective Costs

When multiple metrics are of interest, expressing the cost function  $\Phi_j^{(q)}$  becomes a challenge. A usual approach is to obtain the weighted average of all metrics. However, this approach is not suited for loss or reliability metrics. To

address this limitation, we formulate a multi-objective routing cost  $\Phi_j^{(q)}$  as follows for action  $j$  and CoS  $q$ :

$$\Phi_j^{(q)} = \phi_1 \phi_2 (1 - \beta) + \phi_4 \beta \quad (2)$$

The term  $\phi_1$  establishes the base cost considering the response time. Alternatively,  $\phi_1$  may be given by the hop count  $\phi_0$  instead if needed. The term  $\phi_2$  introduces a loss (or reliability metric) that proportionally magnifies the cost  $\phi_1$ . Additionally, an additive penalty term  $\phi_4$  is included to address miscellaneous costs, such as the monetary expense associated with link usage.

Depending on the selected CoS, certain terms may not be relevant in the cost calculation. As needed, it is possible to set  $\phi_1 = 1$ ,  $\phi_2 = 1$ , or  $\phi_4 = 0$  to nullify their effect. A possible encoding for a CoS field of the bundle is presented in Table I. The approach can be easily extended to include additional metrics of interest. To achieve multiple routing goals, different CoS bits can be set, with their associated costs given by equation (2). We note that all the individual terms in (2) are ratios, i.e., unitless and  $\beta$  is the interest weight factor,  $0 \leq \beta < 1$ , given to the latter cost. This is achieved by normalizing the individual metrics with respect to a reference value.

TABLE I  
INDIVIDUAL ROUTING GOALS AND THEIR POSSIBLE CoS ENCODING.

Label	Encoding	Routing goal
$\phi_0$	000	Path length
$\phi_1$	001	Response time
$\phi_2$	010	Loss
$\phi_4$	100	Monetary cost

Related to  $\phi_4$ , numerous link pricing models exist but they generally fall into two categories: pay-as-you-go or reserved services. In the pay-as-you-go model, the cost is usage-dependent and is often expressed as a rate of dollars per unit time, denoted by  $m_o \rho$ , where  $\rho$  represents link utilization. Reserved services entail a fixed cost independent of usage, expressed as a rate of dollars per second, denoted by  $m_r$ . Typically,  $m_r$  is less than  $m_o$ . While a detailed taxonomy of pricing models is beyond the scope of this study, it is important to note that different pricing models can lead to different routing decisions depending on their associated costs.

The proposed approach provides a framework for simultaneously optimizing multiple objectives in routing decisions that can be directly used to express rewards for a learning agent as the rewards can be expressed as the negative or inverse cost. The precise expressions for the various cost terms are provided in equations (3), (4), (5), and (8). In a later section, these expressions will be applied to a DTN testbed to illustrate their practical use and relevance.

#### B. Analysis of the Individual Costs

The stationary probability distribution  $\pi$  for a DTN link subject to random disruptions was determined in a prior study

[13] using the matrix geometric method. Building on this approach, it is possible to evaluate the theoretical cost of multi-objective routing when the source and destination are connected through  $N$  non-overlapping overlay paths. Specifically, each path can be modeled as a continuous-time Markov chain (CTMC)  $X(t), t \geq 0$  with a two-dimensional state space  $(a, b) : a \in 0, 1, \dots, K; b \in 0, 1$ . Here,  $a$  represents the number of customers in the system, and  $b$  denotes the path state, where  $b = 1$  indicates a disrupted path, and otherwise it is normal. Importantly, this CTMC has a homogeneous quasi-birth-and-death (QBD) process structure, which allows for further analysis and insights into the system's behavior.

In addition to the buffer size  $K$  (which removes the head-of-line bundle only after transmission), the model includes the bundle arrival rate  $\lambda$ , the service rate  $\mu$  (calculated as the reciprocal of the average service time  $S$ ), the average contact duration  $C$ , and the average disruption duration  $V$ . The cost associated with bundle loss ratio  $\phi_j^{(2)}$  through  $j$  is:

$$\phi_j^{(2)} = [(1 - L_j)(1 - l_{i,j})]^{-1} \quad (3)$$

with  $L_j = \pi_K e$  is the overflow probability with  $\pi_K$  the probability of an arrival to a full system and  $e$  a column vector of ones and  $l_{i,j}$  is the channel  $i, j$  corruption probability that depends on the channel bit-error rate and bundle length. Both  $\pi_K$  and  $e$  are of length two as they model the probabilities for both the normal and the disrupted link state. The denominator in (3) gives the probability of successfully delivering a bundle. The loss-related cost  $\phi_j^{(2)}$  is given by the inverse of that quantity.

The cost associated with path length is defined as

$$\phi_j^{(0)} = H_j / H_m \quad (4)$$

where  $H_j$  denotes the average length of the path from node  $j$  to the destination and  $H_m$  represents the maximum path length in the network, which for this basic analysis is just  $\phi_j^{(0)} = 1$ . In contrast, the response time cost  $\phi_j^{(1)}$  is calculated using Little's law, which takes into account the average number of bundles in the system. Specifically, this cost is given by:  $\mathcal{N} = \sum_{n=1}^K n \pi_n e$ :

$$\phi_j^{(1)} = \frac{\mathcal{N}_j}{p_j \lambda_j (1 - L_j)} \quad (5)$$

The normalization factor is assumed to be one. It is worth noting that in practical applications, precise information about many of the model parameters may not be readily available. However, the costs outlined above can still be implemented effectively without such detailed knowledge. For instance, an estimate of  $H_m$  can be obtained based on the average path length across various next-hop options. Similarly, the average response time can be estimated using measurements obtained from the system in question. These practical implementation strategies will be discussed later.

The kind of pricing models determine the network monetary cost. The links associated with the reserved pricing model

contributes to the total cost according to :

$$m_j^{(r)} = \begin{cases} f m_r & \lambda_j (C_j + V_j) < f C_j / S_j, 0 \leq f \leq 1 \\ m_r & otherwise \end{cases} \quad (6)$$

where  $f$  is the fraction of the contact times using reserved pricing. On the other hand, the on-demand pricing model contributes with an amount that depends on their utilization  $\rho_j = 1 - \pi_0 e$ , which gives:

$$m_j^{(o)} = \rho_j m_o \quad (7)$$

It is worth noting that, when it comes to agents making independent routing decisions, only on-demand links are relevant. This is because reserved links incur a constant cost that is independent of the agent's routing decisions. As a result, the agent can assume that  $m_j^{(r)} = 0$ . Cost  $\phi_j^{(4)}$  is then given by:

$$\phi_j^{(4)} = m_j^{(o)} / M \quad (8)$$

where  $M$  is the maximum cost that as in the case of the response time, can be estimated practically thorough observations. If the CoS bit for the monetary cost is not enabled, then this cost is zero in 2. It should be noted that while routing decisions are made based on (8), the assessment of the network monetary cost requires both (6) and (7).

### C. Numerical Evaluation of the Multi-objective Routing Cost

To assess the effectiveness of the proposed multi-objective routing cost expression (2), we examine a scenario where the source agent must choose between two different paths for individual or group of bundles. This simplification to just two options helps to clarify the idea and provide a clear comparison point. Path 1 offers twice the buffer capacity ( $K_1 = 100$ ,  $K_2 = 50$ ) and is twice as fast as Path 2 ( $S_1 = 1$ ,  $S_2 = 1$ ), but it is also twice as costly ( $m_1 = 2$ ,  $m_2 = 1$ ). Additionally, due to random link disruptions Path 1 is available only 30% of the time on average ( $C_1 = 300$ ,  $V_1 = 700$ ), while Path 2 is available 60% of the time ( $C_2 = 600$ ,  $V_2 = 400$ ). The model can predict the long-term performance of the system with a traffic split  $(1 - p) : p$  for the two paths. The optimal choice for  $p$  depends on the metrics of interest and also the traffic flow rate  $\lambda$ . We depict this observation in Figure 1, where we plot the independent variable  $p$  against each of the metrics of interest.

We present two scenarios to illustrate the dynamics and impact of customizing the multi-objective routing function. Figure 2 shows the case where the primary goal is to minimize both the average bundle response time and the monetary cost rate simultaneously. The left panel displays the Pareto front, which is a parametric plot of these two metrics against the hidden variable  $p$ . The right panel shows the values of the proposed multi-objective routing goal for the same two metrics versus  $p$ . Given that the second path is faster, the response time component of the goal tends to favor this path. However, Path 2 is the most expensive of the two, but also the one with a high loss ratio, which tends to reduce the monetary cost. Notably, the scenario does not include bundle loss as

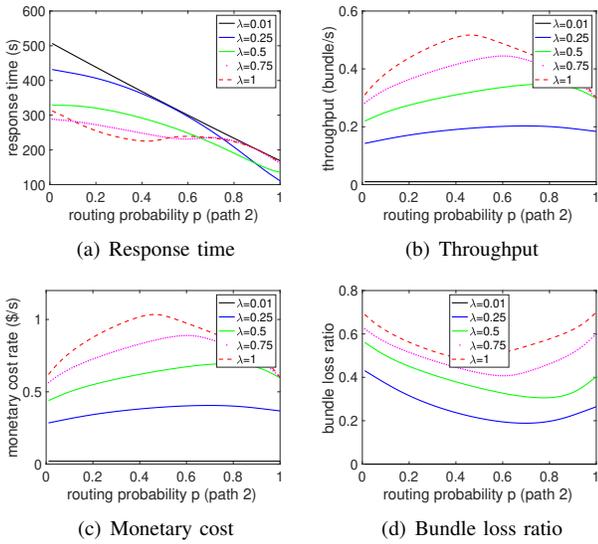


Fig. 1. Flow performance metrics for a case involving two disjoint paths of different characteristics. Model parameters for path 1:  $S_1 = 1$ ,  $C_1 = 300$ ,  $V_1 = 700$ ,  $K_1 = 100$ ,  $m_1 = 2$ . Model parameters for path 2:  $S_2 = 2$ ,  $C_2 = 600$ ,  $V_2 = 400$ ,  $K_2 = 50$ ,  $m_2 = 1$ .

part of the multi-objective target. As a result, the optimal path is Path 2. Figure 3 illustrates the simultaneous optimization of monetary cost and bundle loss. In contrast to the previous scenario, the two objectives are often in conflict. Lower loss can be achieved by splitting the traffic between the two paths, whereas lower monetary cost is best accomplished via Path 2. Thus, the optimal path depends on the traffic rate.

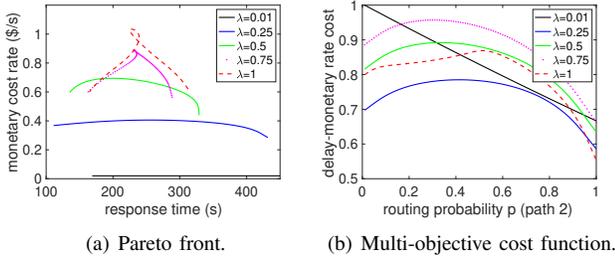


Fig. 2. Pareto front and the proposed multi-objective cost for response time and monetary cost rate with  $\beta = 0.5$ .

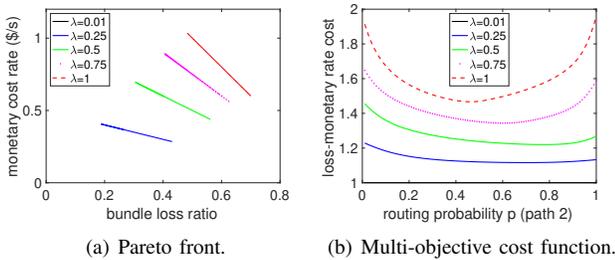


Fig. 3. Pareto front and the proposed multi-objective cost for monetary cost rate and bundle loss with  $\beta = 0.5$ .

#### IV. PROPOSED METHOD FOR MULTI-OBJECTIVE ROUTING

The direct application of analytical results can be challenging due to the need for complete and precise knowledge of the system at the time a new routing decision has to be made. In response to this challenge, a Spiking Neural Network (SNN)-based reinforcement learning method is proposed to maximize the multi-objective function. This method involves an independent agent making routing decisions based on the Cognitive Space Gateway (CSG) method. The CSG represents different outbound port alternatives for a bundle as an SNN structure, which continuously trains its synapse strengths using reinforcement learning and estimated rewards. In this way, it provides a reliable indication of the best outbound port to be used for each bundle. The CSG approach has been shown to be effective in achieving low-delay goals and is extended to achieve multi-objective goals.

A spiking neuron's operation is determined by its membrane potential  $u(t)$  following the Leaky-Integrate-and-Fire (LIF) model:  $\tau \frac{d}{dt} u(t) = -u(t) + RI(t)$ , where  $\tau$ ,  $R$ , and  $I(t)$  are the time constant, leaky resistor value, and input current, respectively. A spike occurs when  $u(t)$  reaches a certain threshold, after which it drops to a refractory rest voltage until recovery. The Cognitive Network Controller (CNC) uses as many excitatory neurons as there are available actions, each spike increasing the receiving neuron's potential. To regulate potential levels, inhibitory neurons connect to all excitatory neurons and their spikes decrease potential levels. The total input current to a neuron is given by  $I(t) = i_e(t) + \sum_f \sum_j \sum_k w_{jk} \cdot i_{jk}^{(f)}(t)$ , where  $i_e(t)$  is the external stimulus and  $i_{jk}^{(f)}(t)$  represents the spike train of unit impulses emitted at times  $f$  by presynaptic neuron  $j$  through the  $k$ -th connection. The CNC determines the routing decision based on the earliest emission of the second excitatory spike. The subsequent two sections highlight the necessary modifications for enabling multi-objective routing with the CSG.

##### Algorithm 1 Bundle routing

---

```

1: procedure BUNDLE_ARRIVAL( $B$ ) ▷ bundle arrives at agent  $x$ 
2:   if node  $x$  is not the destination then
3:     Get end-point from  $B$ : destination  $d$  and CoS
4:      $y = \text{get\_action}(B)$  ▷ from the SNN state
5:      $c = \text{get\_cost}(y, d, \text{CoS})$  ▷ reward  $r=c^{-1}$ 
6:     Apply training step with  $y$ ,  $r$ ,  $d$  and CoS
7:     forward_bundle( $B$ ,  $y$ )
8:   else
9:     Forward  $B$  to upper layer
10:  end if
11:  if node  $x$  is not the source then
12:    Report the local average observations to the predecessor node,
    which include (all zero for  $x = d$ ):
    • Average transmission time to  $d$ :  $\tau_{x,d}$ 
    • Average loss to  $d$ :  $l_{x,d}$ 
    • Average hop count to  $d$ :  $h_{x,d}$ 
    • Average money cost rate to  $d$ :  $m_{x,d}$ 
13:  end if
14: end procedure

```

---

**Algorithm 2** Bundle forwarding

---

```

1: procedure FORWARD_BUNDLE( $B, y$ )  $\triangleright y$  is the link and next-hop node
2:   if buffer  $y$  not full then
3:     Append  $B$  to buffer  $y$ 
4:     Wait until  $B$  reaches the head-of-line
5:     Send  $B$  once possible then remove from buffer
6:     if transmission succeeds then
7:       Measure the service time  $s$ 
8:       Update the averages  $s_{x,y}, \tau_{x,d}, h_{x,d}, m_{x,d}$ 
9:     else
10:      Update the bundle error ratio  $e_{x,y}$  and  $l_{x,d}$ 
11:    end if
12:  else
13:    Drop  $B$ 
14:    Update drop overflow drop ratio  $o_{x,y}, l_{x,d}, m_{x,d}$ 
15:  end if
16: end procedure

```

---

*A. Cost Estimation*

The proposed method for collecting the necessary information is outlined in Algorithm 1–4. Beginning with a new bundle arrival resulting from either a reception from another node or origin from upper layers, Algorithm 1 performs two tasks. For nodes other than the destination, the first task determines the action and cost associated with the outbound link decision. The SNN is then trained for one step before forwarding the bundle to the next hop. The second task is for nodes other than the source and involves sending four pieces of information to the predecessor node. These metrics include the average knowledge of the node’s performance to the destination, i.e., the transmission time, loss ratio, hop count, and money cost rate. The metrics are continually updated using the new bundle transmission observations, as described in Algorithm 2. As buffer overflows and channel losses cannot be entirely eliminated, the agent tracks the average loss observations along with the average service time and hop count, and monetary cost rate. These updates are sent as moving averages to the predecessor node, which replaces the stored values, as described in Algorithm 3.

Algorithm 4 describes how the CSG agent estimates the routing cost as indicated by the bundle’s CoS. If the CoS field’s least significant bit is set to 0, the path length will be used to compute the cost, otherwise, the response time. If the loss ratio bit is set, the estimated loss to the neighbor is computed from overflow and channel loss observations, and then combined by multiplying the cost by the inverse of the success probability. Similarly, if the monetary cost bit is set, the cost rate to the neighbor is computed and added to the total cost by using the  $\beta$  parameter. Only links billed with the on-demand pricing model need to be included, as the reserved model’s cost does not impact routing.

*B. SNN Training*

The regular delay-only reward of the CSG is replaced with the inverse of a cost function (2) to enable multi-criteria routing. This is used to express the reward for forwarding the bundle over the chosen outbound link, with the value being dependent on the required class of service. After selecting

**Algorithm 3** Multi-objective cost estimation

---

```

1: procedure GET_COST( $y, d, \text{CoS}$ )  $\triangleright$  called by agent  $x$ 
2:   if CoS & 0x01 then  $\triangleright$  response time
3:      $\phi = n_{x,y}s_{x,y} + \tau_{y,d} + W_{x,d}$ , where
4:       •  $n_{x,y}$ : buffer occupancy of link  $x, y$ 
5:       •  $W_{x,d}$ : stall time to  $d$  from the contact plan
6:   else  $\triangleright$  hop count
7:      $H = \max(h_{j,d})$ , for all neighbors  $j$ 
8:      $\phi = (1 + h_{y,d})/H$ 
9:   end if
10:  if CoS & 0x02 then  $\triangleright$  loss
11:     $L = 1 - (1 - l_{x,y})(1 - l_{y,d})$ ,
12:    where  $l_{x,y} = 1 - (1 - o_{x,y})(1 - e_{x,y})$ 
13:     $\phi = \phi * \frac{1}{1-L}$ 
14:  end if
15:  if CoS & 0x04 then  $\triangleright$  on-demand model
16:     $m = \text{get monetary cost rate for link } (x, y)$ 
17:    if  $(x, y)$  uses reserved pricing then
18:       $M(x, y) = 0$ 
19:    else
20:       $\rho = \text{get link utilization } (x, y)$ 
21:       $M(x, y) = m\rho$ 
22:    end if
23:     $M = \max(M_{j,d})$ , for all neighbors  $j$ 
24:     $\phi = (1 - \beta)\phi + \beta \frac{M(x,y) + M(y,d)}{M(x,y) + M}$ 
25:  end if
26:  Return  $\phi$ 
27: end procedure

```

---

**Algorithm 4** Report arrival

---

```

1: procedure REPORT_ARRIVAL( $B_r$ )  $\triangleright B_r$  arrives at  $x$  from  $y$ 
2:   Store  $\tau_{y,d}, l_{y,d}, h_{y,d}, m_{y,d}$ 
3: end procedure

```

---

an action and obtaining the multi-criteria reward the agent updates its knowledge of the average cost performance using a moving average formula, which helps it to better adapt to changing conditions. The training step adjusts the SNN weights associated with the last routing decision by an amount proportional to the difference between the latest cost and the minimum value possible with all forwarding options [2].

## V. PROOF OF CONCEPT

To accurately assess the effectiveness of a proposed method, it was put to the test within NASA’s High-Rate Delay Tolerant Networking (HDTN) architecture, a DTN implementation for high-throughput that is compatible with RFC 5050 [3]. The CSG approach described in the previous section was implemented to add multi-objective routing services to HDTN.

A series of experiments were carried out using a network topology depicted in Fig. 4. In this topology, nodes labeled as 100 and 200 were designated as the source and sink of the test traffic, respectively. This particular network topology is representative of a space network, and it presents multiple options for both the shortest and longer paths. Notably, it provides two parallel shortest paths, making it an interesting and relevant network for the experiments. To selected topology was constructed using eight PowerEdge R220 servers, each of which was equipped with multiple network interface ports. To simulate the conditions of independent and full-duplex space channels that handle concurrent bundle transmissions over

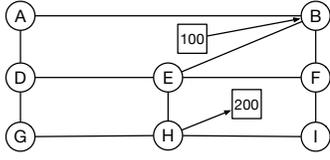


Fig. 4. Laboratory testbed topology. The circles represent extended HDTN [3] nodes with multi-criteria CSG routing and the squares the traffic end-points.

different output ports, we utilized direct Cat 5 twisted pair connections between the servers, without the intermediary of a switch. Additionally, to emulate realistic propagation delays for all physical links, we implemented Linux’s Traffic Control (TC) tool, introducing a 100 ms delay. The use of uniform propagation delays was crucial to simplify the interpretation of the results. We assumed an on-demand pricing model for all links, with the cost rates of 1 \$/s for path B-E-H, 0.4 \$/s for path B-F-I-H, and 0.1 \$/s for all other links.

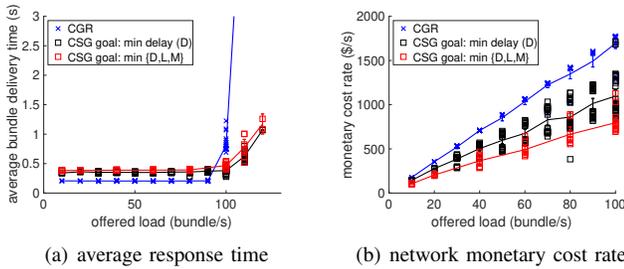


Fig. 5. Performance metrics with multi-objective routing. The baseline results were obtained with CGR.

Additionally, the UDP-based convergence layer adapter was used for the tests with the transmission rate limited to about 100 Bundle/s through the *udpRateBps* value of HDTN with 100 kB bundles. Other relevant parameters include an exponential moving average factor  $\alpha = 0.1$ , a learning rate  $\eta = 0.01$ , and exploration probability  $\epsilon = 0.1$ . By using a standard DTN system, the validity and reliability of the proposed method can be adequately demonstrated.

Fig. 5 shows the average observations for the test flow’s response time and the network monetary cost rate, when the CSG method is assigned either a delay-only routing goal or a multi-objective goal of delay-loss-monetary cost rate. As anticipated, the latter goal achieves higher delay but lower monetary cost compared to the former goal. This outcome provides confirmation of the effectiveness and suitability of the CSG method in achieving multi-objective routing goals. To provide a meaningful reference for the results obtained, we included the results obtained with CGR, which is the standard and sole method available in HDTN. Despite CGR achieving slightly lower delay than CSG for low offered loads, as the reinforcement learning of CSG involves the exploration of sub-optimal paths, the results demonstrate the benefits of CSG over the standard method for both high data rates and multi-criteria routing, thus highlighting its fast convergence to the optimal solution.

## VI. CONCLUSION

In conclusion, this work addresses the challenge of multi-criteria routing in space DTN and other domains. Through our analysis of the system, we identified the complexities of the solution space, which can lead to unexpected results if not properly addressed. To overcome this challenge, we introduced a multi-objective function that effectively balances response time, loss, and generic cost. The monetary cost rate was used as the generic cost metric in this paper.

We evaluated the proposed method by extending the CSG to support routing with multi-objectives and the observation of the required metrics. The laboratory results confirmed that the proposed multi-criteria method can effectively guide reinforcement learning by producing adequate rewards that achieve a suitable trade-off among the metrics of interest. Furthermore, although the evaluation primarily focused on the CSG, the proposed multi-objective formulation is applicable to other DTN routing contexts and domains.

## ACKNOWLEDGMENT

The author gratefully acknowledges the invaluable assistance of the Cognitive Communication Project team at NASA Glenn. This work was supported by grant #80NSSC22K0259 from the National Aeronautics and Space Administration.

## REFERENCES

- [1] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, “Contact graph routing in DTN space networks: overview, enhancements and performance,” *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, mar 2015.
- [2] R. Lent, “A neuromorphic architecture for disruption tolerant networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [3] N. Glenn Research Center, “High speed delay tolerant network,” <https://github.com/nasa/HDTN>. Last retrieved: 12-12-2022.
- [4] S. E. Alaoui and B. Ramamurthy, “Routing optimization for DTN-based space networks using a temporal graph model,” in *2016 IEEE International Conference on Communications (ICC)*. IEEE, may 2016.
- [5] S. Burleigh, C. Caini, J. J. Messina, and M. Rodolfi, “Toward a unified routing framework for delay-tolerant networking,” in *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2016, pp. 82–86.
- [6] M. Marchese and F. Patrone, “E-CGR: Energy-aware contact graph routing over nanosatellite networks,” *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 3, pp. 890–902, sep 2020.
- [7] O. De Jonckère, J. A. Fraire, and S. Burleigh, “Enhanced pathfinding and scalability with shortest-path tree routing for space networks,” in *IEEE International Conference on Communications*. IEEE, May 2023.
- [8] M. Moy, R. Kassouf-Short, N. Kortas, J. Cleveland, B. Tomko, D. Conricode, Y. Kirkpatrick, R. Cardona, B. Heller, and J. Curry, “Contact multigraph routing: Overview and implementation,” in *2023 IEEE Aerospace Conference*. IEEE, Mar. 2023.
- [9] S. Dhara, S. Ghose, and R. Datta, “MFR—a max-flow-based routing for future interplanetary networks,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 5334–5350, dec 2022.
- [10] X. Tian and Z. Zhu, “On the distributed routing and data scheduling in interplanetary networks,” in *ICC 2022 - IEEE International Conference on Communications*. IEEE, may 2022.
- [11] J. A. Fraire and E. L. Gasparini, “Centralized and decentralized routing solutions for present and future space information networks,” *IEEE Network*, vol. 35, no. 4, pp. 110–117, jul 2021.
- [12] S. E. Alaoui and B. Ramamurthy, “MARS: A multi-attribute routing and scheduling algorithm for DTN interplanetary networks,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2065–2076, oct 2020.
- [13] R. Lent, “On the throughput-latency routing optimality of a delay-tolerant network,” in *Globecom*, December 2022.