BSB-Enabled Routing for Deep Space DTNs: Congestion Awareness in Bandwidth-Limited Environments

Evans Tetteh Owusu*, Jonathan Obeng Agyapong*, Theodore Woode*, Clement Heanampong*, Kyoung-Jae Lee‡, Taehoon Kim§, Eunkyung Kim†, and Gosan Noh*
*Dept. of Electronic Engineering, Hanbat National University, Daejeon 34158, South Korea
§Dept. of Computer Engineering, Hanbat National University, Daejeon 34158, South Korea
†Dept. of Artificial Intelligence Software, Hanbat National University, Sejong 30139, South Korea
‡School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, South Korea
{otubrempong, jobengagyapong, woodetheodore, cheanampong}@edu.hanbat.ac.kr
kyoungjae@cau.ac.kr; {thkim, ekim, gsnoh}@hanbat.ac.kr

***Abstract*—Delay tolerant networks (DTNs) for deep space communications lack in-band buffer congestion monitoring, limiting adaptive routing capabilities and buffer management efficiency. We present the buffer status block (BSB), a bundle protocol version 7 extension that enables standardized buffer state reporting through concise binary object representation (CBOR) encoded extension blocks integrated with interplanetary overlay network on delay tolerant network (ION-DTN 4.1.3). We propose BSB-aware deep Q-network (DQN) using deep reinforcement learning to enable adaptive congestion-aware routing from structured buffer state observations in an 8-node bandwidth-constrained topology deployed via containerization. The proposed scheme is benchmarked against contact graph routing (CGR), multi-path replication, and a BSB-Greedy baseline that isolates the contribution of learned routing from BSB information alone. Evaluation across varying workloads demonstrates that BSB-aware DQN achieves the lowest buffer utilization and end-to-end delay through learned congestion-aware routing decisions, while maintaining stable throughput comparable to greedy approaches. Multi-path replication incurs the longest duration due to redundancy overhead, while CGR provides intermediate performance through schedule-based path computation. Together, these results show that the BSB extension enables adaptive routing in unpredictable DTN environments.**

***Index Terms*—Deep space communications, delay tolerant networks (DTNs), deep reinforcement learning, buffer status block (BSB), containerization**

I. Introduction

Deep space communications continue to face fundamental challenges that distinguish them from terrestrial networks: propagation delays spanning minutes to hours, intermittent connectivity due to planetary occlusions, orbital dynamics, and severely constrained bandwidth limited by power and distance [1], [2]. Traditional internet protocols, which assume continuous end-to-end connectivity and rapid acknowledgments, fail in these environments [3]. Delay tolerant networks (DTNs) address these challenges through store-and-forward messaging, where intermediate nodes buffer data during communication outages and forward when links become available [4]. The bundle protocol (BP) standardizes DTN operations for space missions, enabling reliable data delivery across deep space networks despite extreme delays and disruptions [5]. However, BP lacks standardized mechanisms for per-bundle buffer state monitoring, preventing adaptive routing decisions based on network congestion conditions [5]. Current DTN routing protocols operate without visibility into buffer congestion at intermediate nodes [6]. Contact graph routing (CGR), computes paths using pre-configured contact schedules but cannot adapt to dynamic congestion conditions [7], [8]. When relay nodes experience buffer saturation, bundles may be dropped or delayed without the sender's knowledge, leading to inefficient resource utilization. This static approach limits routing efficiency in scenarios with unpredictable traffic patterns, link failures, or when contact schedules become invalid due to operational changes [9]–[12].

Prior work in DTN congestion management has established important foundations. The work in [13] identified buffer utilization as a key congestion indicator and emphasized localized congestion detection, but did not address inter-node buffer information exchange. Fraire et al. [7], [8] proposed congestion-aware CGR variants enhanced Merugu's Floyd-Warshall, cached CGR, path-aware CGR, multi-graph CGR that improve delivery efficiency under deterministic contact schedules, but rely on link-level capacity databases rather than actual node buffer status. Yang et al. [14] applied DQN to learn congestion-aware transmission policies, achieving improved throughput, but focused on link-level rate control without

This work was supported in part by the IITP (Institute for Information & Communications Technology Planning & Evaluation)-ITRC (Information Technology Research Center) grant funded by the Korea government (Ministry of Science and ICT) (IITP-2026-RS-2024-00437886, 40%), in part by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program (IITP-2026-RS-2022-00156212, 40%) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation). This research was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(RS-2025-25408839, 20%).

routing optimization or buffer status exchange. Zhang et al. [15] combined graph neural network (GNN) and reinforcement learning (RL) for adaptive routing in low Earth orbit (LEO) constellations, demonstrating lower delay and adaptability, but operated on simulated models without addressing BP's lack of standardized buffer reporting. These approaches exhibit three limitations: contact-based methods assume static schedules that may become invalid, machine learning approaches still struggle with full inter-node buffer coordination and global congestion visibility, and no existing approach provides standardized per-bundle buffer status exchange during bundle transmission. Furthermore, existing approaches reduce congestion state to a scalar utilization metric, losing visibility into the internal forwarding pipeline stages that distinguish transmission backlog from routing indecision.

To address these limitations, we present the buffer status block (BSB), a bundle protocol version 7 extension that enables per-bundle buffer congestion reporting in DTN. Unlike [14] and [15] which assume buffer state is externally available through simulation oracles, BSB makes structured congestion information a native BPv7 artifact. It encodes a five-metric state vector that distinguishes between transmission queue backlog, unresolved routing decisions (limbo), and active forwarding, enabling identification of the cause of congestion, not merely its presence. We implement BSB in interplanetary overlay network (ION) DTN 4.1.3, the reference implementation for interplanetary networking, and develop a BSB-aware deep Q-network (DQN) routing agent that leverages this structured buffer information to make congestion-aware forwarding decisions.

The main contributions of this paper are summarized as follows: (1) We design a BSB-aware DQN framework that exploits structured multi-metric buffer state for congestion-aware adaptive routing. (2) We propose the first in-band buffer state exchange mechanism for BPv7, implemented as a native extension block within ION-DTN's processing pipeline, eliminating the need for auxiliary management plane signaling. (3) We perform a comparative evaluation against CGR, multi-path replication, and a BSB-Greedy baseline under varying propagation delay in an 8-node Earth-Mars constellation with per-link packet loss and intermittent contacts.

The rest of the paper is organized as follows. Section II presents the system model. Section III formulates the congestion-aware routing problem as a Markov Decision Process. Section IV presents experimental evaluation and comparative analysis. Section V concludes with key findings and future research directions.

II. System Model

A. Network Topology

The experimental testbed implements an eight-node Earth-Mars relay architecture representative of deep space exploration missions, as illustrated in Fig. 1. Each node runs as an isolated docker container with a complete ION-DTN 4.1.3 instance, enabling reproducible multi-node testing without physical hardware. The topology consists of an Earth ground station (Node 1) serving as the data source, six orbital relay satellites (Nodes 2 to 7) providing redundant communication paths, and a Mars surface lander (Node 8) as the final destination. The network exhibits asymmetric bandwidth characteristics typical of deep space communications [16]. The Earth station connects to all relay satellites via high-bandwidth uplinks operating at 50 Kbps, reflecting the transmission capabilities of deep space network (DSN) antennas. The relay satellites maintain 50 Kbps cross-links enabling cooperative forwarding and load balancing. However, all relays face severely bandwidth-constrained downlinks to the Mars lander at 1 Kbps each, creating a persistent bottleneck representative of Ka-band surface links [16]. All inter-node links are subject to one-way light time (OWLT) propagation delay, defined as the time required for an electromagnetic signal to traverse the link distance at the speed of light. Operational Earth-Mars OWLT ranges from 4 to 24 minutes depending on orbital geometry [17]; we apply a compressed OWLT of 1 s and 4 s that preserves the relative delay structure between hops while enabling tractable experimentation within a containerized testbed. Stochastic packet loss is applied per-link at differentiated rates to model the bit-error characteristics of space communication channels [16]: 1% on Earth-relay links reflecting DSN uplink conditions, 0.1% on relay-relay crosslinks representing short-range inter-satellite links, and 2% on relay-Mars surface links capturing the higher error rates of atmospheric Ka-band downlinks. Contact schedules follow intermittent visibility windows governed by orbital mechanics, where relay-to-destination links activate and deactivate according to orbital geometry rather than remaining continuously available.

This topology follows the simplified Mars relay and deep-space DTN scenarios described in the Consultative Committee for Space Data Systems (CCSDS) documentation [18], which model landers, orbital relays, and Earth ground stations connected by asymmetric, bandwidth-limited links with propagation delay and link impairments.

With 8 KB science data bundles, the aggregate downlink capacity of 2 Kbps yields a theoretical maximum delivery rate of 0.09375 bundles per second. When the Earth station transmits at higher rates, this creates sustained congestion at the relay nodes, making buffer management critical to mission success. This bottleneck configuration forces intelligent routing decisions, the core challenge addressed by BSB-aware reinforcement learning.

B. BSB-Aware Routing Framework

Our system integrates three architectural layers to enable congestion-aware routing, as shown in Fig. 2. At the foundation, ION-DTN 4.1.3 provides standards-compliant bundle protocol v7 implementation with CGR, bundle forwarding, and simple data recorder (SDR) buffer management. The BSB extension block handler, integrated into ION-DTN's extension processing pipeline, collects buffer statistics during bundle transmission.

The middle layer implements the BSB protocol, which encodes five critical metrics in CBOR format: total buffer

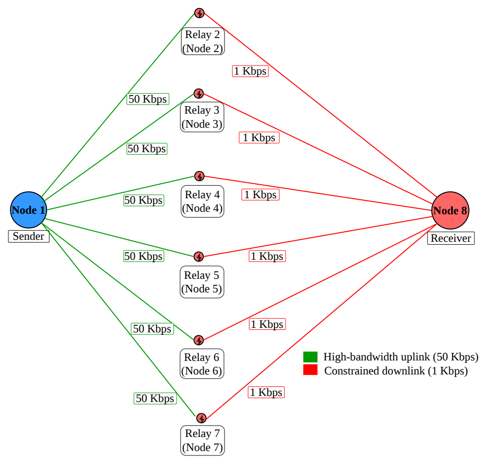

Fig. 1. Earth-Mars relay network architecture showing asymmetric bandwidth allocation.

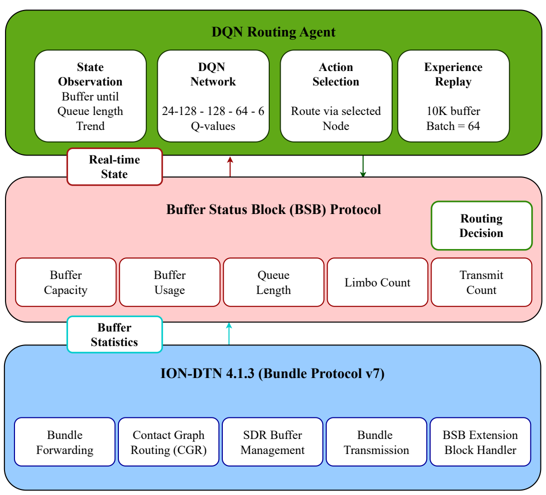

Fig. 2. BSB-aware reinforcement learning routing framework architecture.

capacity, used buffer capacity, queue length (bundles awaiting transmission), limbo count (bundles awaiting routing decisions), and transit count (bundles in forwarding queues). These statistics are attached to outgoing bundles as extension blocks (type 198) and exported to shared memory for immediate access by the routing agent.

The top layer hosts the DQN routing agent, which observes buffer state from the relay nodes, computes Q-values for available routing actions, and issues forwarding decisions to ION-DTN as shown in Fig. 3. This three-layer architecture enables closed-loop congestion control: ION-DTN collects buffer statistics, BSB transmits state information, the DQN agent observes relay congestion, selects optimal routes, and ION-DTN executes bundle forwarding accordingly.

III. Congestion-Aware Routing as Markov Decision Process

We formulate the congestion-aware DTN routing problem as a Markov decision process (MDP) to enable reinforcement learning-based solutions. The MDP framework captures the sequential decision-making nature of routing under buffer constraints, where each forwarding decision affects future network congestion states. The MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is the state space encoding current buffer conditions at relay nodes, $\mathcal{A}$ is the action space representing available routing decisions, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function penalizing congestion and encouraging load balancing, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability function determined by bundle transmission dynamics and network conditions, and $\gamma \in [0, 1]$ is the discount factor balancing immediate and future rewards.

A. State Space

The state space $\mathcal{S} \subset \mathbb{R}^{24}$ captures current buffer conditions at six relay satellites (Nodes 2-7), enabling the routing agent to assess congestion levels across multiple paths. Each state $s_t \in \mathcal{S}$ at time step t is represented as a 24-dimensional vector given by

$$s_t = [u_i, q_i, \Delta q_i, b_i]_{i=2}^{7}, \tag{1}$$

where u_i, q_i, Δq_i, b_i represent buffer utilization, queue length, congestion trend, and used capacity respectively. This multi-relay state representation enables the agent to perform comparative congestion assessment across all available paths, routing bundles to the least congested relay or replicating when multiple paths exhibit favorable conditions.

B. Action Space

The action space $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ consists of six discrete routing decisions available to the agent at each time step where a_j denotes the route bundle via Node $(j+1)$ for $j \in \{1, 2, 3, 4, 5, 6\}$ (single-path forwarding). Each action routes the bundle to one of the six relay nodes (Nodes 2-7), minimizing network overhead by transmitting each bundle once. The agent must select the optimal relay based on current buffer conditions to avoid congestion while maintaining high throughput. The agent selects actions using an ε-greedy policy during training to balance exploration and exploitation. The routing policy can be written by

$$\pi(s_t) = \begin{cases} \text{random action from } \mathcal{A} & \text{with probability } \varepsilon \\ \arg\max_{a \in \mathcal{A}} Q(s_t, a; \theta) & \text{with probability } 1 - \varepsilon, \end{cases} \tag{2}$$

where $Q(s_t, a; \theta)$ is the action-value function approximated by the DQN with parameters θ. The exploration rate ε decays exponentially from 1.0 to 0.01 with decay rate 0.995, gradually shifting from exploration to exploitation as the agent learns effective routing policies.

C. Reward Function

The reward function $\mathcal{R}(s_t, a_t)$ guides the agent toward congestion-aware load balancing while penalizing inefficient resource usage. The reward at time step t is computed based

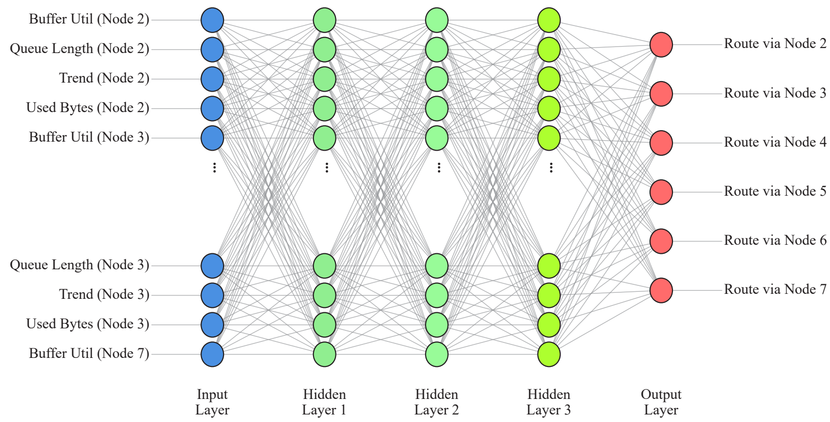

Fig. 3. Deep Q-Network architecture for congestion-aware routing.

on buffer utilization, queue length, and load variance across all relay nodes, which can be given by

$$\mathcal{R}(s_t, a_t) = r_{\text{base}}(u_i) + r_{\text{queue}}(q_i) + r_{\text{balance}}(\text{Var}(\{u_j\}_{j=2}^{7})), \tag{3}$$

where i is the relay node selected by action a_t. The base reward $r_{\text{base}}(u_i)$ encourages routing to underutilized nodes and penalizes routing to congested nodes expressed as

$$r_{\text{base}}(u_i) = \begin{cases} +0.5 & \text{if } u_i \leq 0.5 \text{ (underutilized)} \\ 0.0 & \text{if } 0.5 < u_i \leq 0.7 \text{ (moderate)} \\ -0.5 & \text{if } 0.7 < u_i \leq 0.85 \text{ (high)} \\ -1.0 & \text{if } 0.85 < u_i \leq 0.95 \text{ (critical)} \\ -2.0 & \text{if } u_i > 0.95 \text{ (severe congestion).} \end{cases} \tag{4}$$

The queue length penalty $r_{\text{queue}}(q_i)$ provides additional congestion signal expressed as

$$r_{\text{queue}}(q_i) = \begin{cases} -0.3 & \text{if } q_i > 10 \text{ (high queue backlog)} \\ 0.0 & \text{otherwise.} \end{cases} \tag{5}$$

The load balancing component r_{balance} encourages even distribution across all relay nodes given by

$$r_{\text{balance}}(\text{Var}(\{u_j\}_{j=2}^{7})) = \begin{cases} +0.2 & \text{if Var}(\{u_j\}_{j=2}^{7}) < 0.01 \\ 0.0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\text{Var}(\{u_j\}_{j=2}^{7})$ is the variance of buffer utilization across all six relay nodes, rewarding the agent when load is evenly distributed.

D. Optimization Objective

The agent's goal is to learn a policy π^* that maximizes the expected cumulative discounted reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \right], \tag{7}$$

where $\gamma = 0.99$ is the discount factor that balances immediate and future rewards. The optimal policy is learned by approximating the optimal action-value function $Q^*(s, a)$ using the DQN, which is trained to satisfy the Bellman optimality equation:

$$Q^*(s_t, a_t) = \mathbb{E}_{s_{t+1}} \left[\mathcal{R}(s_t, a_t) + \gamma \max_{a'} Q^*(s_{t+1}, a') \right]. \tag{8}$$

The DQN parameters θ are updated by minimizing the temporal difference error:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2 \right], \tag{9}$$

where $\mathcal{D}$ is the experience replay buffer and θ^- is the target network parameter updated periodically to stabilize training. This formulation enables the agent to learn congestion-aware routing policies that adapt dynamically to network conditions without requiring pre-configured contact schedules or complete network state information.

IV. Experimental Evaluation

TABLE I
Network Simulation Parameters

Parameter	Value
Number of nodes	8
Bundle size	8 KB
SDR heap capacity	1 MB (production/consumption)
Convergence layer	UDP (port 4556)
Earth $\rightarrow$ Relay bandwidth	50 Kbps
Relay $\leftrightarrow$ Relay bandwidth	50 Kbps
Relay $\rightarrow$ Mars bandwidth	1 Kbps (each)
Contact start time	+10 seconds
Contact duration	3600 s (intermittent windows)
Propagation delay (OWLT)	1 s, 4 s
Packet loss (Earth-relay)	1%
Packet loss (relay-relay)	0.1%
Packet loss (relay-Mars)	2%
Bundle generation rate	Variable (20-1500 bundles)

TABLE II
DQN HYPERPARAMETERS

Parameter	Value
Network architecture	24-128-128-64-6
Activation function	ReLU
Optimizer	Adam
Learning rate	0.001
Discount factor (γ)	0.99
Replay buffer size	10,000
Batch size	32
LR Scheduler	StepLR(step_size = 20, gamma = 0.5)
Normalization	BatchNorm1d
Weight initialization	He normal
Weight decay	0.0001 (L_2 regularization)
Target network update	Every 10 episodes
Initial epsilon (ε)	1.0
Final epsilon	0.01
Epsilon decay rate	0.995
Gradient clipping	Max norm 10.0
Training episodes	100
Steps per episode	100

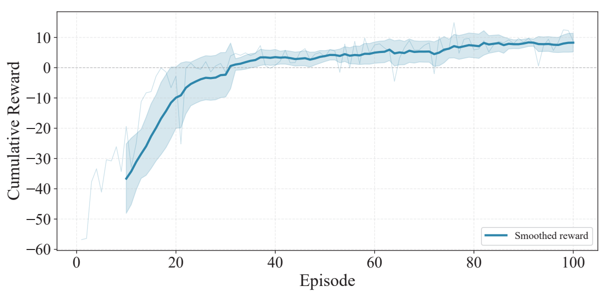

Fig. 4. DQN training convergence over 100 episodes.

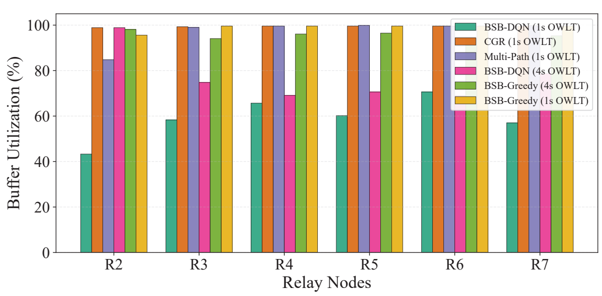

Fig. 5. Buffer utilization across relay nodes for all routing strategies.

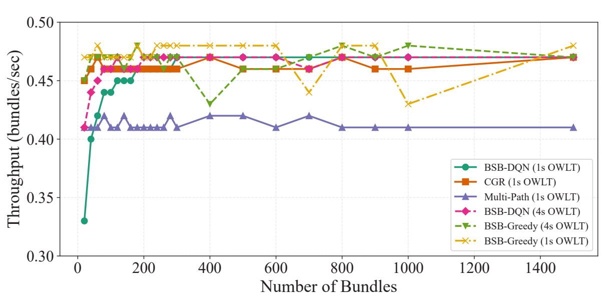

Fig. 6. Throughput comparison across varying workloads.

All experiments are conducted using docker containerization to ensure reproducibility and isolation. Each DTN node runs as an independent docker container with a complete ION-DTN 4.1.3 instance, enabling multi-node testing without physical hardware requirements. The containers communicate over a virtual network (172.20.0.0/24) using user datagram protocol (UDP) convergence layer for bundle protocol v7 transmission. Tables I and II summarize the network simulation parameters and DQN hyperparameters respectively for the 8-node Earth-Mars relay topology.

A. Performance Metrics

We evaluate routing performance using three primary metrics. The throughput τ measures average transmission rate which is given by

$$\tau = \frac{N_{\text{bundles}}}{D_{\text{e2e}}} \quad \text{(bundles/second)}, \tag{10}$$

where N_{bundles} is total bundles transmitted. D_{e2e} is the end-to-end delay and can be obtained by

$$D_{\text{e2e}} = T_{\text{end}} - T_{\text{start}} \quad \text{(seconds)}, \tag{11}$$

where T_{start} is the time when the first bundle of the workload is injected at the source, and T_{end} is the time when the last bundle of that workload is delivered to the destination. The buffer utilization u_i at relay node i quantifies congestion state which is written as

$$u_i = \frac{C_{\text{used},i}}{C_{\text{total},i}} \quad (\%), \tag{12}$$

where $C_{\text{used},i}$ is the occupied buffer capacity, and $C_{\text{total},i}$ is the total buffer capacity at node i.

B. Comparative Results

We compare BSB-aware DQN against CGR, multi-path replication, and BSB-Greedy (selects relay with lowest buffer utilization without learning). To evaluate the effect of propagation delay on BSB-aware routing, BSB-DQN and BSB-Greedy are additionally tested under 4 seconds (s) OWLT.

Fig. 4 shows the training convergence behavior over 100 episodes. The initial negative rewards (-56.8) result from random exploration routing bundles to already-saturated relays, triggering severe congestion penalties. The smoothed reward curve reveals a clear upward trend, with the agent achieving consistently positive rewards after approximately 30 episodes, indicating that the learned policy successfully avoids congested relays and distributes load across available paths.

Fig. 5 presents buffer utilization across relay nodes. BSB-aware DQN under 1 s OWLT maintains the lowest utilization across all relays, demonstrating effective congestion-aware load distribution. BSB-DQN under 4 s OWLT achieves similarly low utilization on most relays with slight elevation on individual nodes attributable to routing decisions based on stale buffer state under higher propagation delay. CGR, multi-path, and BSB-Greedy consistently saturate relay buffers (90–100%) because CGR and multi-path lack per-bundle buffer visibility, while BSB-Greedy lacks a learned policy to exploit buffer information beyond instantaneous greedy selection.

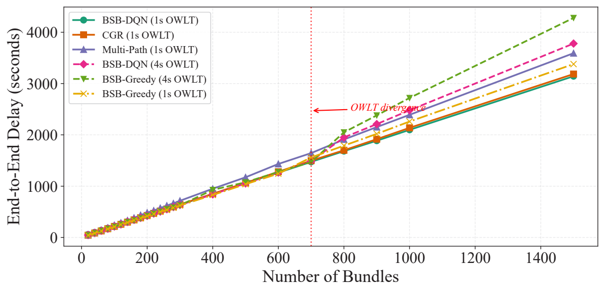

Fig. 7. End-to-end delay comparison. Vertical dashed line marks OWLT divergence at 700 bundles.

Fig. 6 shows throughput against workload size. BSB-Greedy achieves the highest peak throughput due to minimal computational overhead, requiring no model inference per routing decision. However, it exhibits notable oscillation caused by bursty relay switching when multiple relays report similar utilization. BSB-DQN exhibits lower throughput at small workloads due to per-bundle model inference and route computation overhead, but converges to comparable steady-state throughput with significantly lower variance as the overhead amortizes over sustained traffic. CGR approaches BSB-DQN at high workloads while multi-path replication yields the lowest throughput at all load levels, as duplicating each bundle effectively halves useful link capacity.

Fig. 7 plots end-to-end delay against workload size. Under 1 s OWLT, BSB-DQN achieves the lowest delay, slightly below CGR, while BSB-Greedy sits above CGR and multi-path incurs the highest delay due to redundancy overhead. Beyond 700 bundles, the 4 s OWLT variants diverge upward from the 1 s baselines, demonstrating the direct impact of propagation delay on bundle transmission time. BSB-Greedy under 4 s OWLT incurs the highest overall delay, as its reactive policy cannot anticipate congestion when buffer state information is 4 s stale. BSB-DQN under 4 s OWLT outperforms BSB-Greedy despite identical propagation conditions and identical buffer state information, demonstrating that the learned policy extracts more value from BSB data than reactive greedy selection.

V. Conclusion and Future Directions

This paper presented the BSB, a bundle protocol v7 extension that enables per-bundle buffer state reporting in DTNs. We developed a BSB-aware DQN routing agent and compared it against CGR, multi-path replication, and a BSB-Greedy baseline in an 8-node Earth-Mars constellation with per-link packet loss and intermittent contacts. BSB-aware DQN achieves the lowest buffer utilization across relay nodes and maintains stable throughput (up to 0.47 bundles/s), while BSB-Greedy attains marginally higher peak throughput at the cost of higher delay and buffer utilization. Under 4 s OWLT, BSB-DQN outperforms BSB-Greedy despite identical propagation conditions and identical buffer state information, demonstrating that the learned policy extracts more value from BSB data than reactive greedy selection. Future work will extend evaluation to larger topologies with orbital mechanics, integrate BSB with the Licklider transmission protocol (LTP) for reliable delivery, and investigate multi-hop BSB propagation in larger constellations with heterogeneous link characteristics.

References

[1] D. Wang, E. Wang, and R. Wang, "Delay/disruption tolerant networking performance characterization in cislunar relay communication architecture," *Sensors*, vol. 25, no. 1, p. 195, Jan 2025.

[2] R. Besser, "Delay/Disruption Tolerant Networking (DTN) Recommendations Overview," Presentation for Tolerant Networking Group, Dec 2024.

[3] J. Xu, C. Jiang, C. Jiang, N. Ge, and L. Yin, "Space communication network in the new era: Architecture, key technologies, and challenges," *IEEE Netw.*, vol. 37, no. 5, pp. 96–104, Oct 2022.

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," RFC Editor, RFC 4838, Apr 2007.

[5] S. Burleigh, K. Fall, and E. Birrane, "Bundle protocol Version 7," RFC Editor, RFC 9171, Jan 2022.

[6] T. Alhajj and V. Corlay, "Improved contact graph routing in delay tolerant networks with capacity and buffer constraints," *arXiv preprint*, Oct 2024, arXiv:2410.15546.

[7] J. A. Fraire, P. Madoery, and J. M. Finochietto, "Leveraging routing performance and congestion avoidance in predictable delay tolerant networks," in *Proc. IEEE Int. Conf. Wireless Space Extreme Environments (WiSEE)*, Oct 2014, pp. 1–7.

[8] J. A. Fraire, P. Madoery, J. M. Finochietto, and E. J. Birrane, "Congestion modeling and management techniques for predictable disruption tolerant networks," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, Oct 2015, pp. 544–551.

[9] S. Jain and M. Chawla, "Survey of buffer management policies for delay tolerant networks," *J. Eng.*, vol. 2014, no. 3, pp. 117–123, Mar 2014.

[10] H. D. Kim and I. Yeom, "Minimizing the impact of buffer overflow in DTN," in *Proc. Int. Conf. Future Internet Technol. (CFI)*, Jun 2008, p. 20.

[11] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Commun. Surv. Tutor.*, vol. 14, no. 2, pp. 607–640, May 2011.

[12] Y. Zhou, R. Wang, L. Yang, X. Liu, Z. Yang, K. Zhao, and S. C. Burleigh, "A study of link disruption on BP from spatial and temporal perspectives in deep-space vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11 976–11 990, Sep 2021.

[13] P. G. Madoery, J. A. Fraire, and J. M. Finochietto, "Congestion management techniques for disruption-tolerant satellite networks," *Int. J. Satell. Commun. Netw.*, vol. 36, no. 2, pp. 165–178, Mar 2018.

[14] L. Yang, J. A. Fraire, K. Zhao, R. Wang, W. Li, and H. Yang, "Optimizing deep-space DTN congestion control via deep reinforcement learning," *Comput. Netw.*, vol. 255, p. 110865, Dec 2024.

[15] S. Zhang, A. Liu, C. Han, X. Xu, X. Liang, K. An, and Y. Zhang, "GRLR: Routing with graph neural network and reinforcement learning for mega LEO satellite constellations," *IEEE Trans. Veh. Technol.*, Sep 2024.

[16] A. Sabbagh, R. Wang, K. Zhao, and D. Bian, "Bundle protocol over highly asymmetric deep-space channels," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2478–2489, Mar 2017.

[17] G. Kazz, S. C. Burleigh, K.-M. Cheung, and B. Shah, "Evolution of the Mars relay network end-to-end information system in the Mars human era (2030-2040)," in *14th International Conference on Space Operations*, 2016, p. 2420.

[18] Consultative Committee for Space Data Systems, "Rationale, scenarios, and requirements for DTN in space," CCSDS, Informational Report CCSDS 734.0-G-1, Aug 2010, green Book.