Applying the Cognitive Space Gateway to Swarm Topologies

Ricardo Lent
Senior Member, IEEE
University of Houston
Houston, Texas, USA
rlent@uh.edu

Rachel Dudukovich, Adam Gannon, and Robert Short
Communications and Intelligent Systems Division
NASA Glenn Research Center
Cleveland, Ohio, USA
{rachel.m.dudukovich, adam.gannon, and robert.s.short}@nasa.gov

***Abstract*—NASA's future vision for interplanetary networking includes a lunar network, Cube Satellite (CubeSat) constellations, and deep space robotic missions, comprising what could be viewed as a network of networks. Delay-tolerant networking (DTN) architecture and protocols provide a standard network layer among these varying scenarios and mitigate many challenges of the space environment, such as long delays, unplanned service interruptions, and asymmetric links. The Cognitive Space Gateway (CSG) is a routing method in a DTN architecture that uses spiking neural networks as the learning element to optimize routing decisions in a complex environment.**

This work aims to further develop cognitive networking technologies in several critical areas, including DTN, the CSG algorithm, SmallSat swarm topologies, and cloud services. The CSG algorithm is tested in a realistic scenario in which the emulated network topology is based on a SmallSat swarm. The emulation environment will be built upon a commercial cloud service, such as Amazon Web Services (AWS) Elastic Compute Cloud. This work investigates the ability of such a platform to enable a flexible, lower maintenance approach to creating a multi-hop network outside of a physical laboratory. The cloud platform will provide a secure environment allowing for collaboration among government and academic entities.

***Index Terms*—Cognitive Networking, CubeSat Swarms, Routing, Network Modeling, Cloud Computing**

I. Introduction

This paper outlines current efforts to apply the Cognitive Space Gateway (CSG) [1] to future NASA Cube Satellite (CubeSat) and/or SmallSat missions. The CSG is a routing algorithm developed for intermittently connected or delay-tolerant networks (DTN) that uses a spiking neural network to make optimal routing decisions in a multi-hop network. It is envisioned that technologies such as the CSG will enable efficient operation of increasingly complex networking scenarios.

The introduction outlines several candidate NASA missions that may serve as potential demonstrations and infusion paths for the CSG. Section II highlights the technologies related to the CSG and the proposed test environment, as well as how they will enable cognitive networking for NASA's future missions. Section III, Network Modeling, discusses the development of a realistic test scenario for the CSG. Details of the CSG itself are also considered. Section IV, Test Environment, discusses developing a cloud-based simulation environment and the testing approach to further developing the CSG capabilities. Finally, Section V, Future Work, covers the next steps that will be needed to extend the CSG to opportunistic networks.

Potential Infusion Paths

The Cognitive Communications Project at NASA Glenn Research Center (GRC) has identified several upcoming missions that could benefit from cognitive networking technologies such as the CSG. These missions have been analyzed to provide insight into the technology gaps, design considerations, and network conditions that cognitive networking technology attempts to address. These missions include LunaNet [2], HelioSwarm [3], and TechEdSat [4], although there are many other relevant use cases within both the government and commercial sectors. These three missions are discussed briefly to give context to the networking scenario developed in this work.

LunaNet: The NASA Space Communications and Navigation (SCaN) program has been developing the LunaNet architecture, outlining the network infrastructure that will support future missions to the lunar surface [2]. Each LunaNet node will support three standard services: networking services, position, navigation, and timing (PNT) services, and science utilization services. The network services will enable an end-to-end path through surface assets, orbiters, relays, and earth ground stations, transparent to the user. Network operations must be scalable, support interoperability, and perform reliably in a highly mobile, intermittently connected environment.

HelioSwarm: HelioSwarm is a proposed mission consisting of a swarm of nine co-orbiting small satellites (SmallSats) which will study the process of plasma turbulence and is under consideration for NASA's Heliophysics Medium-class Explorer program [3]. Eight "spoke" nodes will transmit science data to one hub, which will forward the data to Earth. Data rates from the individual nodes to the hub will vary with the distance from the hub. Ground stations may communicate with the hub, but will not be able to communicate with the spokes. The hub must be capable of storing the collected data from each node until it has a contact opportunity with the Deep Space Network (DSN).

Technology Educational Satellite (TechEdSat): The TechEdSat program uses a series of multiple CubeSats to evaluate and demonstrate new technologies [4]. These missions are a continuous series of CubeSat demonstrations that are based on a common platform for rapid development. They have demonstrated various technologies, including controlled deorbit and re-entry, wireless sensors, and ISM-band communications links. This rapid prototyping mission concept could be used to test and demonstrate space networking protocols such as the Bundle Protocol [5], Licklider Transmission Protocol [6], and cognitive networking building blocks such as neighbor discovery and link selection. Figure 1 shows an example concept for a potential TechEdsat mission that could be used to demonstrate these foundational technologies.

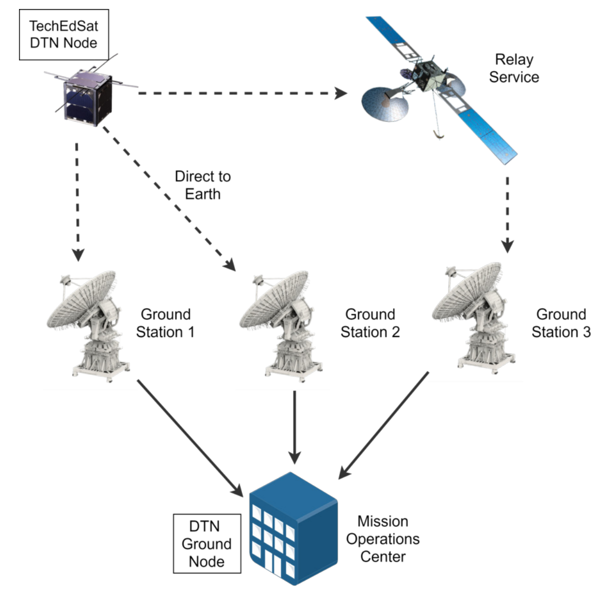

Fig. 1. TechEdSat Networking Mission Concept

II. Cognitive Networking Enabling Technologies

Ivancic et al. define a cognitive network as "having a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions" [7]. Building upon this work and the concepts developed in [8], this section briefly highlights a few of the technologies that may enable the development of cognitive networking within NASA Space Communication and Navigation (SCaN) program.

SmallSat Missions: SmallSat missions will allow for a low-cost way to demonstrate cognitive technologies. An interesting mission concept enabled by SmallSats is that a of distributed system where each satellite performs a portion of a task, and the location of each satellite is used to expand the area covered by a science mission. Crosslinks are needed to enable this distributed sensing environment as well as to perform message relaying between nodes. Related to this is the concept of ad hoc networks where the network topology is changing dynamically as the nodes' distances vary from one another [9]. These distributed missions may benefit from developing multi-agent cognitive systems, such as Multi-Agent Reinforcement Learning (MARL) [10] to enable cooperation between nodes and complex system elements.

Delay Tolerant Networking: DTN can serve as the basis of a cognitive network architecture for space environments as well as other types of mobile, intermittently connected environments. DTN addresses the need for a standard network layer among dissimilar protocol stacks that may exist throughout a single end-to-end path ("user" nodes such as science payloads, relay nodes, gateway nodes, and others). Intermittent connectivity is addressed by using a store-and-forward technique. Protocols such as the Bundle Protocol [5] address the issue of asymmetric link rates by eliminating or reducing the use of acknowledgment-based reliability. Instead, custody transfer requests are embedded within the bundle header. Routing in DTNs is often accomplished at the bundle layer.

Commercial Services: The integration of commercial services into NASA's SCaN program will increase flexibility in the number of communication options science missions will have access to [8]. NASA has plans to pursue commercialization of near Earth Direct-To-Earth (DTE) communications and identify implementation steps to enable commercial DTE services by 2023 [11]. Several commercial ground station services already exist, including Amazon Web Services (AWS) Ground Station, Kongsberg Satellite Services, Atlas Space Operations, and Infostellar.

Artificial Intelligence and Machine Learning: The advances in Artificial Intelligence (AI) and Machine Learning (ML) can facilitate the management of increasingly complex networks. As SCaN evolves from NASA-managed services to incorporate commercial services and develops new network architectures such as LunaNet, networking capabilities must address scalability, reconfigurability, and system-level autonomy. AI and ML can be applied to performance optimization, such as maximized throughput, minimized energy consumption, and monetary cost, as well as fault detection and recovery.

III. Network Modeling

To test the effectiveness of the CSG routing algorithm in a swarm environment, we examined several SmallSat and CubeSat swarm missions including: Starling [9], HelioSwarm [3], and Starlink [12]. We considered realistic scenarios based on the number of nodes, network topology, contact schedules, data rates, data volume, and node storage and processing capabilities. Ultimately, an example LEO constellation roughly based on the Starlink mission was used as a guide to develop the network model. We used approximate orbits and reasonable radio parameters to establish the connections to be used in the test. The simulation was then used to provide input data (contact start time, stop time, and average distance between nodes) for the network emulation environment.

Network Configuration

SpaceX Starlink is an example of a commercial mega-constellation, with the initial LEO constellations planned to consist of 4425 satellites by 2024 [13]. The satellites operate in the Ku (12-18 GHz) and Ka (26.5-40 GHz) bands and intend to provide broadband communication services to worldwide customers [12]. Orbits of spacecraft in the example network are generated from a subset of satellites that comprise the Starlink constellation. This example network consists of sixty-one satellites with fifty-four forming the main belt and seven in crossing orbits. In this scenario, we allow any satellite to connect to the six ground stations we selected across the United States and consider RF inter-satellite links between satellites in the constellation. This network model is designed to test how the CSG scales with the size. For the initial work in this paper, we selected a subset of 12 nodes from the model to create the test environment for the CSG algorithm.

Figure 2 shows the larger sample network of sixty-one satellites and six ground stations within the Satellite Orbit Analysis Program (SOAP) [14] that is used to determine the satellite orbits and line-of-sight between network assets.

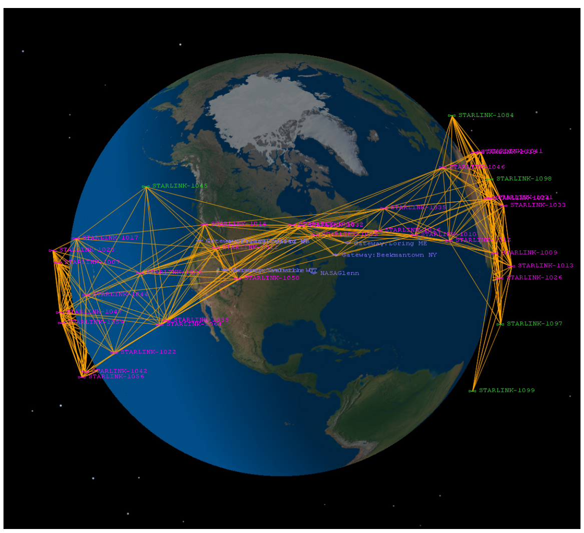

Fig. 2. Satellite Orbit Analysis Program Visualization[14]

Cognitive Space Gateway

The Cognitive Space Gateway (CSG) [15, 1] will be used to make optimized routing decisions for multi-hop network scenarios such as the multi-satellite missions outlined in the previous section. The CSG uses reward shaping to determine suitable rewards for reinforcement learning and a spiking neural network as the outbound link decision-making element for data bundles. Prior work demonstrated that CSG achieves improved performance when compared to techniques such as Contact Graph Routing (CGR) [16].

Such advantage is achieved through the iterative learning of the best decisions through rewards that include the expected network-wide conditions (e.g., congestion). Additionally, the Cognitive Network Controller, the basis of the CSG, has been demonstrated on the SCaN Testbed on-board the International Space Station (ISS), earning it a higher technology readiness level (TRL) than many proposed cognitive networking technologies [1]. For these reasons, this paper seeks to continue developing and testing the CSG to prepare it for additional flight missions.

IV. Test Environment

The CSG will be tested with a cloud-based network of containers using AWS. The benefit of this approach is that multiple users from both NASA and academia will be able to access the test environment remotely. In addition, the approach is a meager cost in comparison to purchasing and configuring local servers, and the system will be very flexible in terms of cloning additional container-based nodes for extensive emulations such as the Starlink inspired scenario. A custom network simulator was developed in conjunction with the CSG for integration into the system. This network simulation environment will allow for algorithm and software development independent of a hardware testbed. Once the algorithms are determined to perform satisfactorily, they can be integrated into the software components of a realistic, software-defined radio testbed. This approach will allow algorithm development to be completed in parallel and independently from lower-level protocol development.

Cloud Computing

Figure 3 shows a simple network simulation developed using AWS Elastic Compute Cloud (EC2). Each node is an EC2 instance that is part of a Virtual Private Cloud (VPC) developed for the Cognitive Communications Project. Each node is based on an Ubuntu 18.04 Amazon Machine Image (AMI). These container instances represent a SmallSat or ground station within the simulation. The CSG software will be installed within the Linux environment and a custom network simulator controls the network interfaces of the containers, according to the network models developed in the previous section.

The initial test setup is simple to allow for the initial configuration of the EC2 instances, network settings, and CSG software. Once the initial nodes have been configured, replicating nodes for increasingly complex network configurations should be a simple task. Instance parameters such as memory, processors, and storage options are being evaluated to determine the best performance of the system while minimizing cost.

Test Network

The test network consists of 12 EC2 *t2.micro* (1 CPU, 1 GB RAM) instances that were connected according to the selected network model (see Fig. 4). The *t2.micro* instances are limited to one or two network interfaces, so IPIP tunnels were established to recreate the required topology where nodes can have 8 or 11 bi-directional network ports. With IPIP, separated (virtual) interfaces were created to handle the packet transmissions over each link. This approach allows the

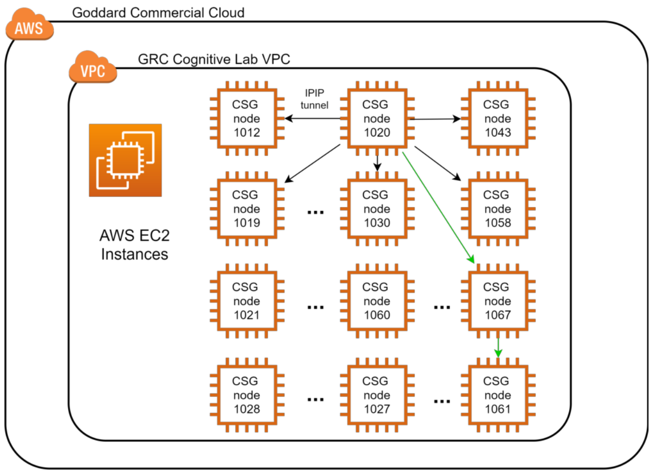

Fig. 3. Cognitive Networking Emulation Environment in EC2

emulation of the link features as required by the network model through the application of packet buffering and filtering on a per-interface basis.

Because of the large number of nodes of the test network, many communication paths may not require more than one link. To evaluate the effectiveness of the routing adaptation, two arbitrary and disconnected nodes (i.e., requiring at least 2 hops) were selected as the source and sink for a test bundle flow. For reference performance, the path offering the earliest time of arrival (ETA) was also calculated from the contact graph, which is the method used by the Contact Graph Routing (CGR) protocol. The algorithm finds the shortest contact path by considering the waiting time for contacts. The input includes the link rates and propagation delays associated with each contact so that the shortest ETA can be found. However, the impact of dynamic characteristics of the links, such as the network-wide buffer occupancies and random packet losses are not included in the computation, as they are generally unknown, which may impact the optimality of the results.

V. Performance Measurements

The network is assumed to be connected for the entire duration of the test traffic using nodes 1020 and 1061 as the source and sink. This is a reasonable assumption given the large path redundancy that exists between any two nodes and the fact that only four of the 57 network links are affected by link disruptions according to the network model. The flow consists of 1,000 bundles of 100 kB each and the fastest route considering the earliest time of arrival (ETA) but without taking into account any possible network congestion is 2 hops long (e.g., via node 1067). The propagation delay of the links is given as calculated by the satellite orbit simulation and it is assumed that all links provide negligible packet loss rates, except for link 1067–1061 that is assumed to be affected by large signal loss yielding a packet loss ratio of 0.02. This consideration was purposely introduced to make the routing problem non-trivial. The affected link sits on the shortest ETA path. Each single-hop transmission was handled by a custom

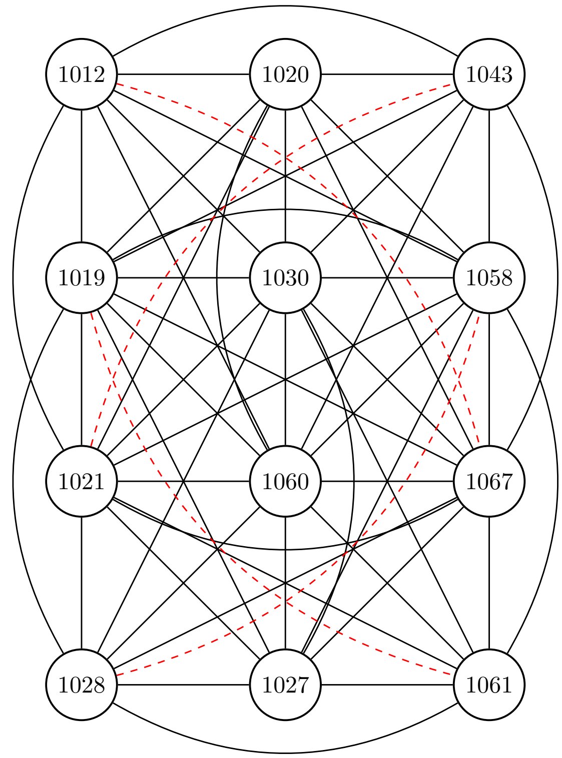

Fig. 4. Test network topology.

implementation of the Licklider Transmission Protocol (LTP) configured to manage 100% red-part blocks with up to four concurrent sessions and a maximum segment payload size of 1,300 bytes. Because LTP achieves bundle transmission reliability by implementing selected segment retransmissions, the single-hop bundle delivery time tends to increase when packet losses occur on the link 1067–1061.

Time-series Observation

Fig. 5 (a) shows the time series of the bundle delay to the sink (i.e., the *response time*) that was observed for each bundle in a sample run. In this test, the bundles were sent at the rate of 10 bundle/s. The segment retransmissions that are required to reliably deliver bundles over link 1067–1061 contribute to increasing the end-to-end bundle latency as shown in the figure given that the buffer occupancy along the shortest ETA path builds up quickly with the incoming packets. The CSG method uses path latency estimations calculated from single-hop performance metrics that allow modifying the synapse weights of the SNN and achieve continual learning. This feature is demonstrated by the lower bundle latency that is achieved with the CSG approach compared to the shortest ETA attempt. Because the CSG uses exploration, which is a necessary mechanism for learning, a certain percentage of the bundles may experience larger delays than the rest. The delay peaks that can be observed with the CSG occur as a result of the use of the high-packet loss link or due to the intra-flow network congestion. In the latter case, the delay is caused by the wait time for other bundles that already occupy the buffers waiting to be transmitted. It can be observed that the peaks tend to decrease over time, which is an indication of the learning effectiveness of the system to achieve low end-to-end bundle latency.

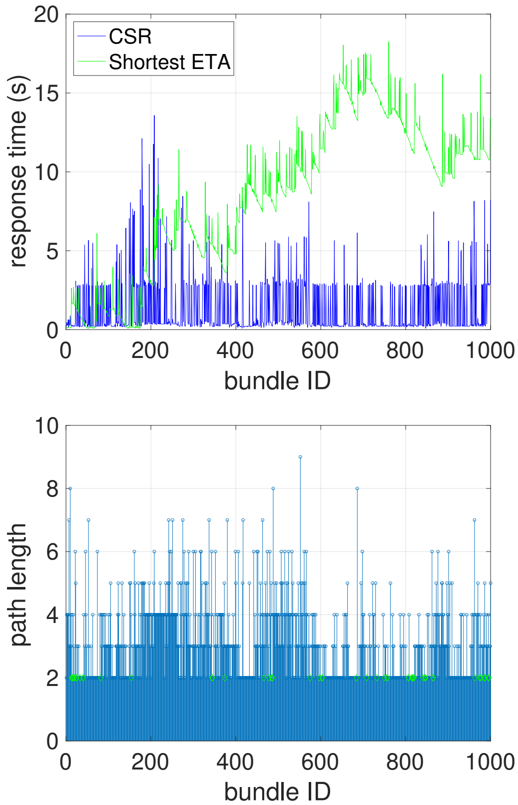

Fig. 5. Time series of (a) the bundle delivery latency and (b) path length of bundles of a sample test flow. In the former case, results obtained both with the CSG and the shortest ETA are shown. For (b), only the path length with CSG is shown and the instances using the affected link have been highlighted using green diamond-shaped markers.

It is worth noting that multiple 2-hop paths are possible between the source and sink with the selected topology, but only one of them was affected by the larger packet loss ratio in the experiments. No attempts were made to balance the traffic among the 2-hop paths in the reference case, whereas a load-balancing effect over paths of any size results as a side-effect of the CSG. For the same experiment, Fig. 5 (b) depicts the path length taken by each consecutive bundle. With CSG, the path affected by the higher loss was used about 5% of the time and those instances have been highlighted in the figure using green diamond-shaped markers. The low usage count for this path suggests that the CSG learned to avoid that route.

Average Trends

To obtain the average trends, multiple and identical episodes of the flow transmission were registered. The average response times of the bundles using the traffic sending rate as the experimental factor are shown in Fig. 6. With light traffic, the CSG performs slightly worse than the shortest ETA because of the overhead brought by the path exploration. However, as the traffic load increases, the reference method becomes severely penalized while the CSG is capable of achieving bundle response times that are appreciably lower.

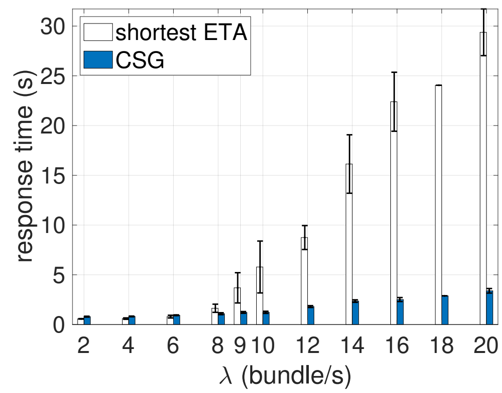

Fig. 6. Average bundle response time as a function of the flow sending rate.

While the bundle response time has been the unique routing objective used to decide the paths in this study, it is interesting to observe the end-to-end throughput that results from that assumption. As can be observed in Fig. 7, the results are correlated with the bundle response time results. With low-traffic rates, both methods achieve similar performance, but once network congestion becomes a major factor for higher sending rates, the CSG achieves better throughput than the reference method.

It is worth observing how the CSG moves its route selection preferences to longer paths after a workload level increase despite such paths accumulate larger propagation delays. Fig. 8 contrasts the distribution of the path length selection using 4 and 12 bundle/s workloads. The path length grows about 7% from 3.0011 with light traffic to 3.2130 with heavy traffic. These values provide further indication of the route adaptation capabilities with the CSG learning in real-time how to use longer but less congested paths as needed.

VI. Future Work

In conclusion of this paper, there are several areas that have been identified for future work. The first area is to further evolve the CSG Python source into flight-like software suitable for potential SmallSat missions. We will investigate existing DTN implementations and software frameworks to determine the next step towards integrating the CSG algorithm. Additionally, several areas will be investigated to develop cognitive networking technologies further.

More complex network scenarios can be developed for the emulation environment. An increasingly dynamic network topology and/or a larger number of nodes will necessitate a more opportunistic style of routing. The CSG will require additional development to achieve opportunistic routing capabilities. These improvements will increase the number of

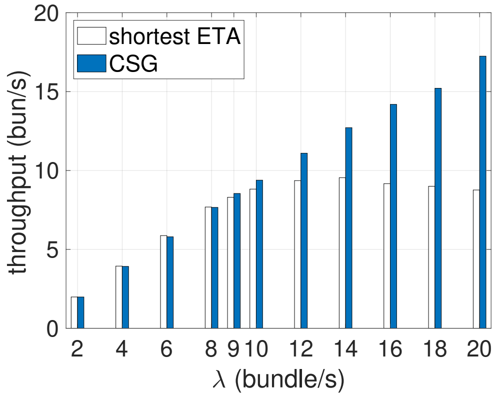

Fig. 7. Throughput vs. flow sending rate.

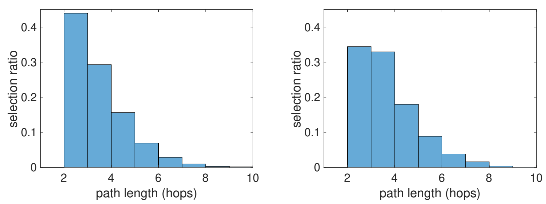

Fig. 8. Observed selection of path lengths across all collected samples with sending rates of (a) 4 bundle/s and (b) 12 bundle/s.

missions the CSG can be applied to and increase the flexibility, robustness, and reconfigurability of the system.

Additional work is also needed to enable neighbor node discovery, which will be essential for opportunistic style contacts. Several approaches to discovery exist, including Link Layer Discovery Protocol (LLDP) and DTN IP Neighbor Discovery [17] [18].

ACKNOWLEDGMENTS

The authors wish to acknowledge the contributions of Dr. Janette Briones, Dave Chelmins, and Jason Mitchell for their continuing support of the Cognitive Communications Project at NASA GRC. We also wish to acknowledge the Mission Cloud Platform team at Goddard Space Flight Center for their efforts in provisioning and configuring cloud computing resources for this work. Dr. Lent was supported by the grant #80NSSC17K0525 from NASA's Space Technology Research Grants Program.

REFERENCES

[1] R. Lent, D. Brooks, and G. Clark. "Validating the Cognitive Network Controller on NASA's SCaN Testbed". In: *2020 IEEE International Conference on Communications (ICC)*. Dublin, IE, June 2020.

[2] D. Israel et al. "LunaNet: a Flexible and Extensible Lunar Exploration Communications and Navigation Infrastructure and the Inclusion of SmallSat Platforms". In: *the 34th Annual AIAA/USU Virtual Conference on Small Satellites*. 2020.

[3] L. Plice, A. Dono Perez, and S. West. "HelioSwarm: Swarm Mission Design in High Altitude Orbit for Heliophysics". In: *AAS/AIAA Astrodynamics Specialist Conference*. 2019.

[4] M. Murbach et al. "The TechEdSat-N Series: A Collaborative Technology Development Platform in the Nano-Satellite Form Factor". In: *International Space Station Research and Development Conference*. 2018.

[5] K. Scott and S. Burleigh. *Bundle Protocol Specification*. https://tools.ietf.org/html/rfc5050. 2007.

[6] M. Ramadas, S. Burleigh, and S. Farrell. *Licklider Transmission Protocol-Specification, RFC 5326*. https://tools.ietf.org/html/rfc5326. 2008.

[7] W. Ivancic et al. *Cognitive Networking With Regards to NASA's Space Communication and Navigation Program*. Tech. rep. NASA Glenn Research Center, 2013.

[8] D. Chelmins et al. "Cognitive Communications for NASA Space Systems". In: *International Communications Satellite Systems Conference (ICSSC)*. 2019.

[9] H. Sanchez et al. "Starling 1: Swarm Technology Demonstration". In: *32nd Annual AIAA/USU Conference on Small Satellites*. Logan, UT, USA, 2018.

[10] K. Zhang, Z. Yang, and T. Basar. "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms". In: *CoRR* abs/1911.10635 (2019). arXiv: 1911.10635.

[11] G. Heckler. *SCaN Commercialization of Near-Earth Communications Services - Strategic Overview*. https://www.nasa.gov/sites/default/files/atoms/files/scan_commercialization_nac_202101016_final3.pdf. 2013.

[12] G. Giambene, S. Kota, and P. Pillai. "Satellite-5G Integration: A Network Perspective". In: *IEEE Network* 32.5 (2018), pp. 25–31.

[13] Federal Communications Commission. *Space Exploration Holdings, LLC, Application for Approval for Orbital Deployment and Operating Authority for the SpaceX NGSO Satellite System*. https://www.fcc.gov/document/fcc-authorizes-spacex-provide-broadband-satellite-services. 2018.

[14] D. Y. Stodden and G. D. Galasso. "Space System Visualization and Analysis using the Satellite Orbit Analysis Program (SOAP)". In: *Proceeding of the 1995 Aerospace Applications Conference*. Mar. 1995, pp. 369–387.

[15] R. Lent. "A Neuromorphic Architecture for Disruption Tolerant Networks". In: *2019 IEEE Global Communications Conference (GLOBECOM)*. Dec. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013605.

[16] S. Burleigh. *Contact Graph Routing*. https://tools.ietf.org/html/draft-burleigh-dtnrg-cgr-00. 2009.

[17] M. Rodolfi. "DTN Discovery and Routing: From Space Applications to Terrestrial Networks". MA thesis. University of Bologna, 2014.

[18] D. Ellard, R. Altmann, and A. Gladd. *DTN IP Neighbor Discovery (IPND)*. https://tools.ietf.org/html/draft-irtf-dtnrg-ipnd-02. Nov. 2012.