

Network Emulation Testbed Capabilities for Prototyping Space DTN Software and Protocols

Ronny L. Bull*, Rachel M. Dudukovich[†], Juan A. Fraire[‡], Nadia Kortas[†], Robert Kassouf-Short[†], Aaron Smith[†],
Ethan Schweinsberg[†]

*SUNY Polytechnic Institute, College of Engineering, Department of Computer Science, Marcy, NY, USA.

[†]NASA Glenn Research Center, Cleveland, OH, USA.

[‡]Inria/INSA Lyon, Villeurbanne, France - CONICET-UNC, Córdoba, Argentina.

Abstract—This paper delves into network emulation tools essential for evaluating and designing Delay Tolerant Networking (DTN) protocols in space and satellite networking. It surveys and assesses the capability of current testbeds to create realistic test environments crucial for developing and evaluating DTN protocols. Specifically, this study provides a comprehensive overview of key DTN protocol stacks and related network emulation platforms and a detailed exploration of NASA’s research facilities. Finally, the paper underscores the fundamental emulation capabilities and the importance of a standardized framework for scenario creation, highlighting its vital role in evaluating and advancing future emulation platforms for space DTN.

Index Terms—Space and Satellite Networking, Delay/Disruption Tolerant Networking (DTN), Bundle Protocol (BPv7), Network Emulation and Simulation

I. INTRODUCTION

Network emulation tools for Delay Tolerant Networking (DTN) [1] create realistic environments for developing, testing, and evaluating DTN protocols, routing strategies, and applications. These tools can replicate various conditions in challenging and unpredictable networking environments, including high delays, intermittent connectivity, and limited bandwidth. Delay-tolerant networking is a specialized communication technology useful when traditional network assumptions, such as low latency and continuous connectivity, do not apply. Some common use cases for DTN include deep-space communications, aerial networks, underwater sensor networks, and intermittently connected networks in remote areas or disaster-stricken regions. Realistic scenarios can be created, emulated, replayed, and modified using network emulation tools, making them invaluable for designing and optimizing DTN-based communication systems before deployment in production environments.

This paper provides an overview of space-centric DTN software and applicable network emulation platforms, defines a standard set of emulation testbed capability requirements,

and suggests guidelines for scenario development. Our work provides an introduction to DTN emulation. It can be used as a starting point for evaluating these platforms for developing and prototyping DTN software suites, protocols, and applications.

II. DTN SOFTWARE SUITES

This section provides a detailed comparison of major DTN stacks, highlighting their distinct capabilities and features. DTN provides store-and-forward capability for disrupted links, and central to this functionality is the Bundle Protocol (BP). The protocol defines the required data units and procedures for developing networked space communication links. Table I, adapted and extended from Internet Society Interplanetary Chapter[2] Pilot Projects Working Group’s analysis[3], lists various technical features available across DTN implementations. Each column represents a specific DTN stack, allowing for a side-by-side comparison of their support for different versions of BP: BPv6[4] or BPv7[5]. Additionally, it addresses core functionalities such as Convergence layer Adapters (CLAs), which interface with underlying protocols like TCP, UDP, and LTP, as well as BPsec security extensions. The table also covers custody transfers, essential for responsibility transfer and congestion control, alongside Contact Graph Routing (CGR)[6][7], a preferred routing method in multi-hop space DTNs. The table also evaluates integration with key protocols such as the Consultative Committee for Space Data Systems (CCSDS) Space Packet Protocol (SSP) for encapsulation, Bundle Streaming Service Protocol (BSSP) for streaming, Asynchronous Message Service (AMS) for management, and CCSDS File Delivery Protocol (CFDP) for file transfer.

This comprehensive assessment offers a complete picture of each DTN stack’s capabilities and underscores the diversity of DTN implementations. Furthermore, it highlights the critical need for specialized testbeds to assess and ensure the interoperability of these diverse systems, a vital aspect in the effective deployment of DTN technologies.

III. SIMULATION VS. EMULATION

Network simulation and emulation are techniques used to test and evaluate network systems and protocols. However, there are some key differences between the two. Network simulation is primarily used for modeling and analyzing

This manuscript is a work of the United States Government authored as part of the official duties of employee(s) of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All other rights are reserved by the United States Government. Any publisher accepting this manuscript for publication acknowledges that the United States Government retains a non-exclusive, irrevocable, worldwide license to prepare derivative works, publish, or reproduce the published form of this manuscript, or allow others to do so, for United States government purposes.

TABLE I: DTN Stacks and Features

Feature/Stack - Subfeature	ION	IONE	HDTN	uD3TN	DTNME	CFS	Unibo	IBR
BPv6	Y	Y	Y	Y	Y	Y	N	Y
BPv6 - TCPCLv3	Y	Y	Y	Y	Y	-	N	Y
BPv6 - UDPCL	Y	Y	Y	Y	Y	-	N	Y
BPv6 - LTPv1	Y	Y	Y	Y	Y	-	N	N
BPv6 - BPSEC	Y	Y	N	N	N	-	N	Y
BPv6 - Custody BPv6	Y	Y	Y	Y	Y	-	N	N
BPv7	Y	Y	Y	Y	Y	Y	Y	N
BPv7 - TCPCLv3	Y	Y	Y	Y	Y	-	Y	N
BPv7 - TCPCLv4	Y	Y	Y	Y	Y	-	N	N
BPv7 - UDPCL	Y	Y	Y	Y	Y	-	N	N
BPv7 - LTPv1	Y	Y	Y	Y	Y	-	Y	N
BPv7 - BPSEC	Y	Y	Y	N	N	-	N	N
BPv7 - Custody (with BIBE)	Y	Y	N	Y	Y	-	N	N
BPv7 - RTP	N	N	Y	N	N	-	N	N
CGR, SABR	Y	Y	Y	N	N	-	Y	Y
CCSDS SPP	N	N	N	Y	N	-	N	N
BSSP	Y	Y	N	N	N	-	N	N
AMS	Y	Y	N	N	N	-	N	N
IPv6 (for CLAs)	N	Y	Y	N	N	-	Y	Y
IPND	Y	Y	N	N	N	-	N	Y
CFDP	Y	Y	N	Y	Y	-	N	N
Language	C	C	C++	C	C++	C	C++	C++

network behavior in a controlled and synthetic environment. It aims to simulate network events, interactions, and behaviors to understand how a network or protocol would perform under various conditions. Network emulation focuses on replicating real-world network conditions and events as closely as possible, providing an environment that mimics a specific network or network component's behavior, including real hardware and software.

There are advantages and disadvantages to network simulation versus emulation for DTNs. Discrete event simulators such as OMNeT++[8] and custom-developed simulators[9] can evaluate operational scenarios covering a several-day period much faster than in real-time. This is often necessary for space scenarios where satellites in LEO may take 90 minutes or longer to complete one orbit around the Earth. During algorithm and prototype development, it can be helpful to abstract away lower-level details to focus on modeling high-level behavior and allow multiple test cases to be rapidly evaluated. Simulation tools may also focus more on intuitive visualizations that help demonstrate messaging between nodes, orbital patterns, and node mobility.

Several simulation platforms are available for use with DTN, such as the ONE simulator[10], DtnSim[11], and the ESTNet[12], as well as several purpose-built custom implementations, as surveyed in [13]. Besides these specialized simulators, various general network simulation platforms, such as NS3[14] and OMNeT++[15] exist, although these do not possess all the features necessary for DTN simulation out-of-the-box. However, it is worth noting that proposed mechanisms[16] for integrating DTN into OMNeT++, DtnSim, and ESTNet were specifically developed to integrate with OMNeT++, leveraging its event-driven simulation scheduler to model communication between DTN nodes. On the other hand, the ONE simulator was explicitly developed as a stan-

dalone network simulator focusing on DTN.

In contrast, there are several use cases where discrete event simulators may not be sufficient. Algorithms and protocols that rely on state information and feedback across multiple layers of the communication system, as well as metadata based on realistic interaction between nodes, may be difficult to simulate beyond a basic level[17],[18],[19]. While existing code from real implementations can be ported to simulation models (e.g., DtnSim[11] interfaces with ION and HDTN libraries in accelerated simulations), container-based emulations and hardware-in-the-loop can make this porting process unnecessary and will also produce more accurate results. In such cases, scenario times can often be scaled to reduce execution time without significantly affecting the emulation's validity. In preparation for flight experiments, high-fidelity software verification and validation must use emulation techniques incorporating as much of the real flight system or accurate engineering models as possible. They will often be executed in real-time[20].

Most developers will likely use a combination of both simulation and emulation as the software lifecycle progresses[21]. Simulation tools support rapid and experimental development and can also serve as a visualization front-end. Containers and hardware platforms can be networked with delay emulators[22],[23],[24], and propagation simulators[25] for high fidelity, cross-layer testing and development. Additional tools and applications must often be developed to visualize the network and interactions between nodes[26].

IV. NETWORK EMULATION PLATFORMS

Many network emulators are available, all with different capabilities and levels of fidelity [13]. This section provides an overview of the network emulation platforms that have proven applicable to DTN applications.

A. Software-based Emulation Tools

1) *CORE\EMANE*: The Common Open Research Emulator (CORE) and Extendable Mobile Ad-hoc Network Emulator (EMANE) are network emulation tools developed by the U.S. Naval Research Laboratory[27][28]. CORE uses Linux containers and namespaces to emulate nodes, along with Linux bridging and virtual interfaces to connect the virtualized nodes. CORE provides a GUI interface, a Python API, distributed execution, and hardware-in-the-loop capabilities. It is best suited to emulate OSI layer 3 and above. EMANE can be interfaced with a CORE scenario for realistic layer modeling for layers 1 and 2. Users can define custom EMANE radio models for a higher fidelity emulation, although CORE also provides a basic wireless interface model. Both tools are open source and freely available to the public.

2) *EXata and Systems Tool Kit*: EXata Network Modeling software is maintained by Keysight[29] and is focused on developing network digital twins. EXata has both simulation and emulation modes and a 3-D, navigable scenario viewer. EXata provides hardware-in-the-loop, human-in-the-loop capabilities, extensive metrics generation, and reporting, as well as interfaces to tools such as Wireshark[30] and Systems Tool Kit (STK)[31]. EXata is focused on layers 3 and above but can also model layers 1 and 2. Real and virtual nodes can be mapped to an EXata emulation at layer 2. STK can be used for a detailed analysis of the physical layer. Radios can be created in an STK scenario and interfaced with EXata for network layer simulation. Ansys maintains STK and provides a wide variety of digital twin capabilities. Both tools require a paid, yearly license.

3) *VirtualBricks*[32]: is a virtualization tool for managing Qemu or KVM virtual machines and network devices, specifically tailored for creating DTN testbeds in research. It provides an interface to facilitate the configuration of these virtualized testbeds, enabling easy management of virtual machines and network design. Virtualbricks' application extends beyond research, aiding in novel routing evaluations[33] and Mars-Earth data transfers[34]. Its ease of use also makes it valuable in educational settings, simplifying testbed and network design for learners.

4) *EmuStack*[35]: provides a robust network emulation platform for DTNs, leveraging OpenStack, Docker, and network services like Linux traffic control. It facilitates precise time synchronization, efficient resource management, and high-performance emulation control, enabling the operation of multiple virtual nodes on a single physical node. Notably, EmuStack achieves substantial efficiency, with a setup of 60 virtual nodes incurring only about nine percent CPU usage.

5) *SPICE*: The SPICE testbed[36] is pivotal for satellite and space communications research. It uniquely incorporates real and flight-ready components, including HellasSat 2 satellite access, to facilitate authentic testing conditions. SPICE enables interoperability tests across DTN implementations like ION, DTN2, and IBR-DTN. Its application is extensive, ranging from evaluating routing algorithms[37] and erasure

coding in space DTNs[38] to analyzing the DTN transport layer performance in ION using the LTP protocol[39].

6) *TUNIE*: The Tsinghua University Network Innovation Environment (TUNIE)[40] is a virtualized DTN emulation testbed. Utilizing XEN Hypervisor for running virtual machines as DTN nodes and OpenFlow for link event management, TUNIE integrates the DTN2 protocol. It supports custom routing and resource allocation schemes. TUNIE's focus extends beyond individual DTN implementations to the overall performance evaluation of DTNs, offering various mobility scenarios to assess metrics like transmission efficiency and scalability.

7) *Cloud Platforms*: Cloud-based platforms can be used to create networks of virtual machines that can be used to emulate DTN scenarios. One benefit over the previously mentioned tools is that cloud containers are easily accessed by multiple users from different organizations and are hosted globally. However, the user would need to develop custom scripts to emulate a disconnected network and provide their own user interface. Cloud platforms also provide images that emulate many operating systems and hardware architectures and can provide specialized processors such as advanced graphics processing units (GPUs). Several papers discuss DTN emulation in cloud environments such as CloudLab[41], Amazon Elastic Compute Cloud[18], and federated systems across Google Cloud, Amazon Web Services (AWS) and Microsoft Azure[42]. CloudLab is free for researchers, and commercial services like AWS provide a free tier of less powerful containers. However, larger computing resources require payment.

B. Research Facility Testbeds

This section discusses research facilities home to networks and physical testbeds focused on wireless network emulation. Three of the testbeds discussed are part of NASA Glenn's Aerospace Communications Facility[43], which is among the top NASA facilities for radio frequency communications technology research and development.

1) *NASA Delay-Tolerant Networking Engineering Network*: The DTN Engineering Network (DEN)[21] (Figure 1) consists of labs at NASA Centers and occasionally external partners, connected through VPNs and virtualization servers managed at NASA Glenn Research Center. Current users include NASA Goddard Space Flight Center, Marshall Space Flight Center, Jet Propulsion Laboratory, and Glenn Research Center. DTN interoperability testing has been conducted between HDTN, DTNME, BPLib, and ION using the DEN[20]. Virtual machines hosted on the DEN can use a Netropy 10G4[23] network emulator to add delays, configurable bandwidth, and other impairments such as jitter, packet reordering, and packet loss. Internal network connections support up to 10 Gbps; however, external connections over VPN operate at lower rates. The DEN was also recently connected to NASA Goddard's Mission Cloud Platform, which can enable larger, multi-region networks.

2) *Glenn Research Center DTN Automated Test Suite*: Glenn Research Center has implemented a DTN Automated

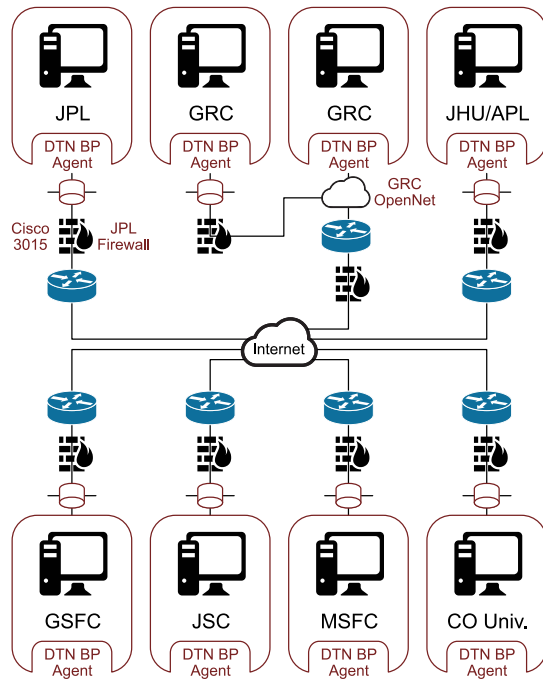


Fig. 1: NASA Multi-Center DTN Engineering Network (DEN)

Test Suite to deploy and test multi-node DTN scenarios easily in the High Rate Delay Tolerant (HDTN) laboratory. The Netropy network emulation device introduces parameters such as long delay, loss, and corruption between space and earth nodes. Tests were scheduled via Gitlab CI/CD and stored in the GRC Gitlab Repository. They include the runscripts, configuration files, and JSON files defining the Netropy parameters such as latency, bandwidth, link, and data disruptions. Containers are deployed for each node. Docker swarm was used to orchestrate applications on the network where one machine acts as the “swarm manager” (Figure 2) and is responsible for deploying, removing, and monitoring applications. This facilitates the automated deployment of complicated scenarios to the network. After a test, containers are removed from each node, and each machine writes logs and data to an NFS. The swarm manager copies and processes the data, and the results, including logs, stats, and generated plots, are pushed to GitLab for easy access.

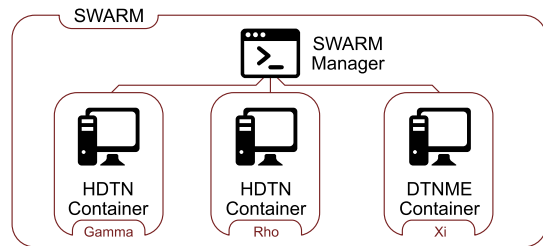


Fig. 2: Node Orchestration using swarm.

This test suite was used to emulate cislunar scenarios (Figure 3) and Geostationary scenarios to prepare for HDTN

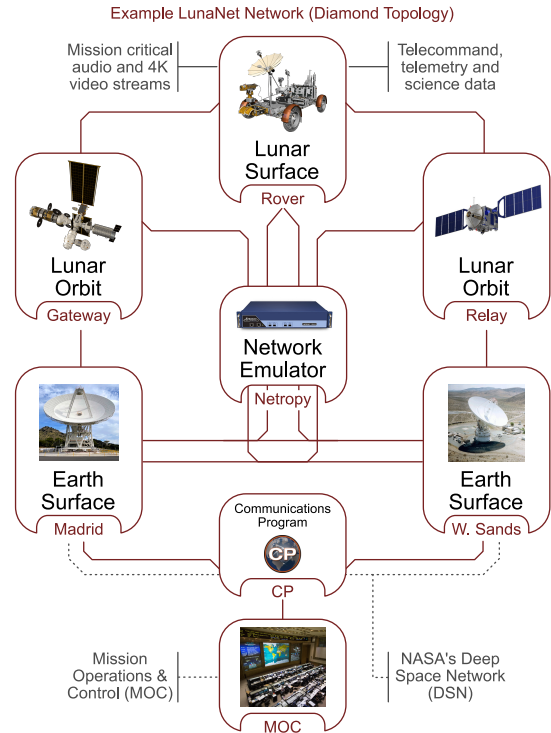


Fig. 3: Cislunar Scenario Emulation in the GRC DTN Automated Test Suite.

infusion, such as International Space Station (ISS) upgrades to support enhanced operations and laser communications using Integrated LCRD Low-Earth Orbit User Modem and Amplifier Terminal.

3) *NASA Cognitive Ground Testbed and MATRICS*: The Cognitive Ground Testbed[44] emulates a broad suite of communication functionality in satellite missions. These include data generation and storage management, service scheduling, RF transmission and reception, and networking. The tests typically involve LEO spacecraft that generate science data onboard the spacecraft and transmit to Earth via relay satellites during ground station passes. The tests run in real-time and incorporate multiple software-defined radios (SDRs), flight computers, a channel emulator, commercial modems, and an array of software for orbital analysis, mission operations, and networking. The testbed manages a simulated world to determine potential contact intervals and the RF environment of connected radio pairs. The test scenarios[9] include definitions of the satellite orbits, ground station locations, transmitter and receiver parameters, radio waveforms, and data generation profiles for the spacecraft. The simulated environment builds on the System Tool Kit (STK) software and provides RF channel parameters—delay, Doppler offset, and attenuation—to the Keysight PropSim F64 Radio Channel Emulator. During a test, flight hardware generates simulated mission science data and autonomously schedules upcoming ground station passes with the service providers. During the scheduled contacts, the satellite radio chooses an appropriate

waveform for the service provider. It transmits HDTN bundles over an RF channel at carrier frequency through the channel emulator and to the commercial modem. The defining features of this testbed are its high-fidelity emulation of the entire data lifecycle from generation to delivery, the incorporation of storage management and scheduling, and the ability to evaluate multiple simultaneous missions. Combining these features enables a holistic evaluation of the communication infrastructure's capacity and resilience.

The MATRICS testbed[45] is also being developed at Glenn Research Center. MATRICS utilizes a 25-ft anechoic chamber to measure over-the-air Ka-band transmissions. Spacecraft and/or terminal orientation can be electronically manipulated during a test using a robotic arm. This testbed capability enables research in next generation Ka-band relay and user terminal solutions, mitigates pre-flight risks, and provides NASA with a high-fidelity assessments of user terminals. The capabilities of the Cognitive Ground Testbed will be merged with MATRICS in the newly constructed Aerospace Communications Facility at Glenn Research Center.

4) *JSC Software Development and Integration Laboratory:* The Software Development and Integration Laboratory (SDIL) at Johnson Space Center performs software testing, network integration, and flight image preparation of the ISS DTN onboard gateway, payloads, and experiments[46]. The SDIL uses virtual machine images identical to those onboard the ISS, including flight and ground software packages for file transfer, DTN traffic, and other administration tools. The Ku-band link from space to ground is emulated using a Linktropy network emulator. The SDIL can also connect to the Huntsville Operations Support Center to test end-to-end data flows using the real ISS ground network. The SDIL has tested ION, DTNME, and HDTN for integration into the ISS Joint Station LAN for flight experiments and operations[20].

V. NETWORK EMULATION CAPABILITIES REQUIRED FOR SPACE APPLICATIONS

Evaluating the emulation platform's effectiveness at providing the correct fidelity level for the specific needs of testing DTN technologies for a project or mission is very different than evaluating the DTN technology itself. Some common metrics are typically considered when testing DTN software suites and protocols. However, it is important to understand that these metrics measure the capabilities of the DTN software or protocols being tested, not those of the emulation platform on which the DTN is running. Most relevant performance metrics are listed in Table II.

When evaluating network emulation platforms for DTN applications, it is necessary to consider the previously mentioned metrics, as the system needs to emulate an environment that can effectively push the limits of the software and protocols being evaluated to allow for meaningful data to be collected. The data produced will depend upon the capabilities of the tested DTN software suite and the level of detail of log verbosity when active. Therefore, the emulation environment must possess the proper capabilities to replicate the intended

TABLE II: DTN Performance Metrics

<i>Delivery Ratio:</i> Percentage of messages successfully delivered to the final destination node. Fundamental goal in DTNs.
<i>Latency:</i> Time for message delivery is typically higher in DTNs due to store carry-and-forward. Critical optimization parameter.
<i>Message Overhead:</i> Additional in-packet data used by DTN protocol to manage bundles. Low overhead is a must for constrained systems.
<i>Protocol Overhead:</i> Overhead originating from extra control messages for acknowledgments, routing, and management purposes.
<i>Buffer Usage:</i> Memory used for message storage and forwarding. A key parameter for resource-constrained space nodes.
<i>Routing Efficiency:</i> Message routing efficiency is measured by path length, hops, and energy consumed to deliver a given bundle.
<i>Message Drop Rate:</i> Rate of message loss during transmission. Reducing the drop rate is crucial to ensure efficiency and dependability.
<i>Contact Durations:</i> The duration of contacts is the limiting factor in message forwarding capacity in DTNs.
<i>Data Delivery Time:</i> Time for successful message delivery, including latency and routing efficiency.
<i>Resource Utilization:</i> Use of bandwidth, energy, and storage. Essential in remote or constrained environments.
<i>Security and Reliability Metrics:</i> Data integrity, authentication, network robustness against disruptions or attacks.
<i>Scalability:</i> Performance as the DTN network scales in nodes, time horizon, and message traffic.
<i>Fairness:</i> Evaluation of resource allocation equality among nodes/users (storage, energy, computing power, etc.).

production environment, which, at minimum, must include the ability to orchestrate node movement, skew and maintain timing relative to the mission being emulated, modify link connectivity, bandwidth, and throughput across the entire emulated network, and introduce planned and unplanned delays and disruption into the DTN being tested.

Higher fidelity levels can be achieved by introducing topology visualization, custom radio modeling, and hardware-in-the-loop capabilities, which can lead to a higher Technical Readiness Level (TRL)[47] of the DTN application being developed or tested. Beyond basic radio and network modeling, high-fidelity emulations may include elements of the real physical layer (optical or RF over-the-air testing or wired laboratory connections), ground segment components, flight-like hardware, and detailed verification and validation of adherence to protocol standards via automated black-box and white-box tests.

Fig. 4 shows graphical interfaces from four main emulation tools studied in this paper. These include:

- 1) a widely validated commercial simulation and emulation environment (EXata and STK, Fig. 4-a));
- 2) an open and freely available simulation and emulation alternative (CORE, Fig. 4-b));
- 3) a 3D visualization tool to gain intuition on the scenario scale and dynamics (Unity 3D Tool, Fig. 4-c));
- 4) a DTN stack testbed used for software verification and validation to prepare for deployment in space (GRC HDTN Automated Test Suite, Fig. 4-d)).

Access to freely available tools such as CORE ensures that preliminary studies can be achieved in a low cost manner, allowing small businesses and academia to conduct tests in their own environments. Commercial tools such as STK and

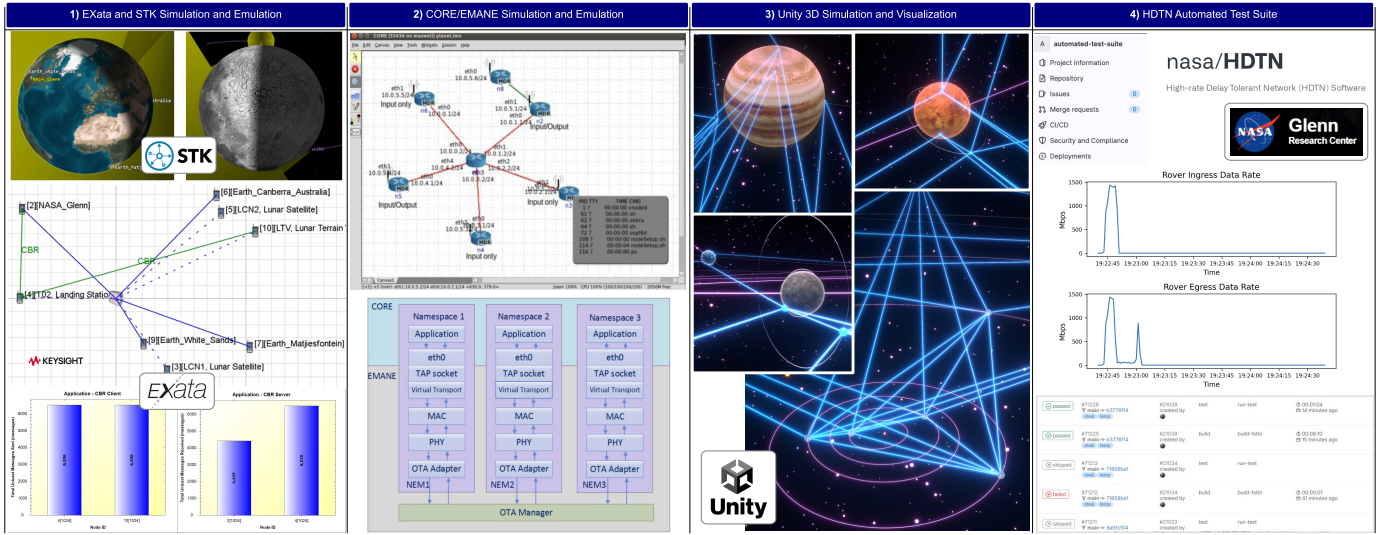


Fig. 4: Selected tools for simulation and emulation demonstration: a) EXata and STK commercial tool, b) CORE/EMANE open-source tool, c) Unity 3D Visualization tool, and d) HDTN automated test and deployment suite.

EXata may be used by industry and government, providing product reliability and customer support.

VI. RESULTS

In this section we showcase some of the test results achieved in several of the testbeds and emulation platforms discussed in this paper. Since this paper is a study and comparison of different capabilities, the results are not an exhaustive explanation of each test. In depth results of the various tests are the subject of future work or include a reference to a completed publication.

A. Bundle Protocol Priority Testing on the Cognitive Ground Testbed

The Cognitive Ground Testbed was used to model a user spacecraft generating data with two levels of priority based on the Bundle Protocol version 6 priority field. A script was used to generate data files of each priority at specified intervals. The technologies being tested were HDTN's bundle storage priority management and User Initiated Services scheduling capabilities. Contacts to the ground must be scheduled such that both high and low priority data are delivered to the mission operations center by the specified deadline. HDTN's storage module must ensure that high priority bundles are queued for transmission before low priority bundles. Fig. 5 shows that high priority bundles (priority 2) interrupt the transmission of low priority bundles (priority 1) once they have been created. Transmission of low priority data resumes once all of the high priority data has been received. Fig. 6 shows that high priority data is delivered with a lower latency than low priority data.

B. Cognitive Routing Tested in CORE

The Cognitive Ground Testbed is focused on using realistic flight-like radios and RF channels in a LEO scenario. As such, future technologies at a lower TRL such as cognitive

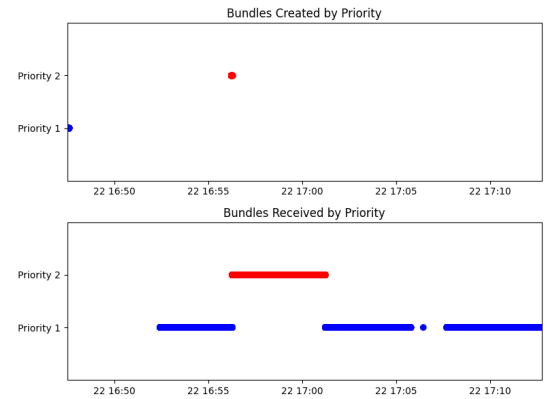


Fig. 5: Bundle Priority Testing in the Cognitive Ground Testbed

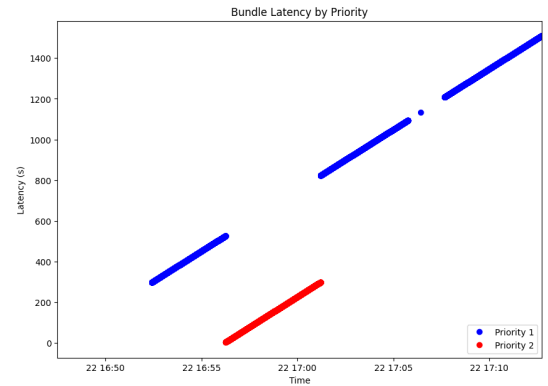


Fig. 6: Bundle Latency According to Priority

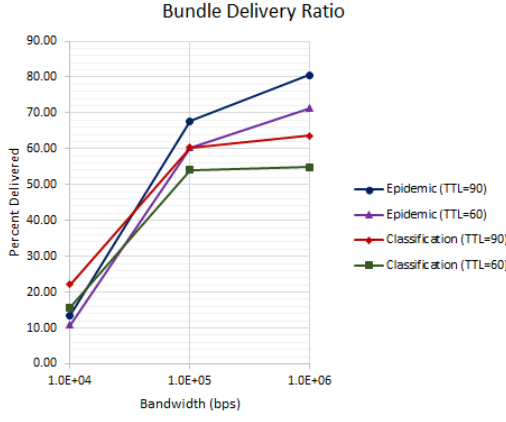


Fig. 7: Bundle Delivery Ratio using CORE

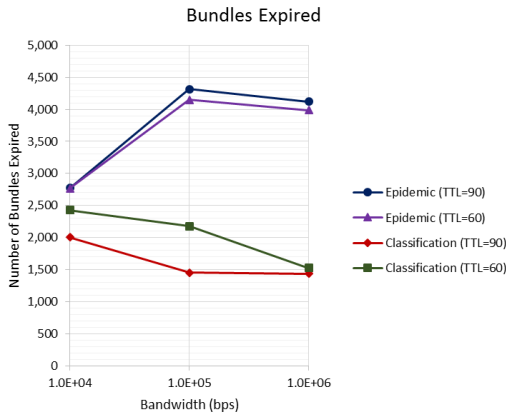


Fig. 8: Bundle Expiration using CORE

routing are better suited to be tested in a network emulation environment such as CORE. CORE has been used to evaluate cognitive routing techniques in ION, HDTN, and IBR-DTN [48]. Epidemic routing was compared to cognitive routing based on a machine learning classification technique. The scenario consisted of 20 nodes following several different mobility models. Nodes transmit data at specified intervals with varying bundle time-to-live. Fig. 7 shows the bundle delivery ratio for epidemic routing versus the classification approach. Fig. 8 shows that fewer bundles expire using the classification approach.

C. Bundle Protocol Security Testing in the GRC Automated Test Suite

The GRC automated test suite was used to test the impact of enabling transport layer security on data rates by sending files from the rover to the MOC with and without Bundle Protocol Security (BPsec) enabled in an emulated cislunar scenario shown in Figure 3. Depending on orbital mechanics, the rover can have a direct line of sight to a single node or to multiple nodes. The Netropy network emulator was used to introduce a lunar delay of 1500 ms between the earth and space

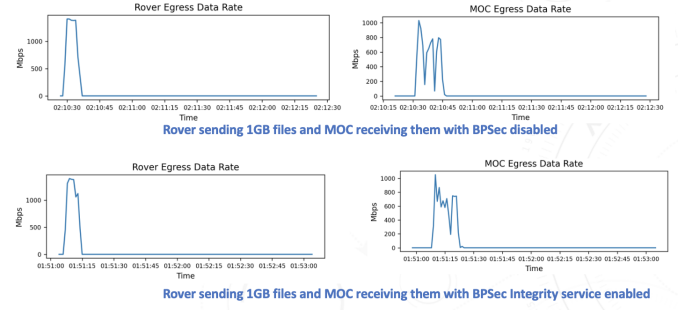


Fig. 9: Data rates results with and without security in GRC automated test suite Cislunar scenario

nodes. Figure 9 shows that the sending and receiving nodes achieved similar data rates and that BPsec did not degrade the performance despite the long delays and large files being sent. In addition, it confirmed that high rates in the order of multi-gigabits were still possible with BPsec enabled in a realistic mission scenario.

VII. FUTURE WORK

In conclusion, this paper discussed several aspects of selecting a network emulation environment for DTN prototype testing. Our team recommends maintaining several tools based on the specifics of the use-case including: TRL of the technology to be tested, organizations requiring access to the test environment, ease of collaboration, analysis and reporting capabilities, visualization capabilities, and ease of use.

Our future work aims to develop a standard set of scenarios that represent the production mission. The scenarios should be developed as generic templates independent of any platform to be adapted to run on emulation solutions with various capabilities, from the simplest to the most complex. These scenario templates should include a detailed written description, a network diagram, and a contact plan delivered in a standardized format[49]. The template serves as a blueprint that allows a developer or researcher to create an adaptation of the scenario specific to each emulation platform to be evaluated. Additionally, the scenario templates facilitate qualitative and quantitative comparisons among the testbeds discussed in this paper and provide a foundation for developing future emulation platforms. This way, the scenario remains consistent where the emulation processes, capabilities, and fidelity may change drastically between platforms. A future goal is to create a set of standard guidelines for DTN scenario development that can transcend all testbed environments, facilitating the proliferation of open-sourced scenarios that developers can use when prototyping.

VIII. ACKNOWLEDGMENT

We would like to acknowledge that the descriptions of simulation platforms (ONE, DtnSim, ESTNet) and emulation tools (VirtualBricks, EmuStack, and SPICE) presented in this paper were adapted from Tobias Nöthlich's Master thesis at the Technical University of Dresden (TUD) [13].

REFERENCES

- [1] V. Cerf, S. Burleigh, L. Torgerson, R. Durst, K. Scott, K. Fall and H. Weiss. Delay-Tolerant Networking Architecture. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc4838>
- [2] IPNSIG. Internet Society Interplanetary Chapter. [Online]. Available: <https://ipnsig.org/>
- [3] IPNSIG PWG, "Bundle protocol implementations," <https://ipnsig-pwg.github.io/>, 2023, accessed: 2023-12-27.
- [4] K. Scott, S. Burleigh. Bundle Protocol Version 6. [Online]. Available: <https://datatracker.ietf.org/doc/rfc5050/>
- [5] S. Burleigh, K. Fall and E. Birrane. Bundle Protocol Version 7. [Online]. Available: <https://datatracker.ietf.org/doc/rfc9171/>
- [6] G. Araniti and N. Bezirgiannidis and E. Birrane and I. Bisio and S. Burleigh and C. Caini. Contact graph routing in DTN space networks: overview, enhancements and performance. [Online]. Available: <https://ieeexplore.ieee.org/document/7060480>
- [7] J. A. Fraire, O. De Jonckère, and S. C. Burleigh, "Routing in the space internet: A contact graph routing tutorial," *Journal of Network and Computer Applications*, vol. 174, p. 102884, 2021.
- [8] OMNeT++ Discrete Event Simulator. [Online]. Available: <https://omnetpp.org/>
- [9] A. Smith, E. W. Brown, and F. Merat, "A digital testbed for autonomous spacecraft communication services," in *2023 IEEE Space Hardware and Radio Conference*, 2023, pp. 20–22.
- [10] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Simutools '09*, Brussels, Belgium, 03 2009. [Online]. Available: <https://doi.org/10.4108/ICST.SIMUTOOLS2009.5674>
- [11] J. A. Fraire, P. Madoery, F. Raverta, J. M. Finochietto, and R. Velazco, "Dtnsim: Bridging the gap between simulation and implementation of space-terrestrial dtns," in *SMC-IT 2017*. IEEE, 2017, pp. 120–123.
- [12] A. Freimann, M. Dierkes, T. Petermann, C. Liman, F. Kempf, and K. Schilling, "Estnet: a discrete event simulator for space-terrestrial networks," *CEAS Space Journal*, vol. 13, no. 1, pp. 39–49, 1 2021. [Online]. Available: <https://doi.org/10.1007/s12567-020-00316-6>
- [13] T. Nöthlich, "Comparative analysis of delay-tolerant networking protocol implementations: Towards a unified performance assessment framework," Master's thesis, Faculty of Computer Science, TU Dresden, Germany, April 2024.
- [14] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.
- [15] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Simutools '08*, Brussels, Belgium, 03 2008.
- [16] O. R. Helgason and K. V. Jónsson, "Opportunistic networking in omnet++," in *Simutools '08*, Brussels, Belgium, 2008.
- [17] M. Moore, R. Bull, S. Burleigh, J. Cook, J. Waszkiewicz, D. Cook, and J. Seif, "Reactive routing: Harnessing advanced software modems," in *CCAAW'23: IEEE Cognitive Comms. for Aerospace Applications Workshop*, 2023.
- [18] R. Lent, R. Dudukovich, A. Gannon, and R. Short, "Applying the cognitive space gateway to swarm topologies," in *2021 IEEE Cognitive Comms. for Aerospace Applications Workshop (CCAAW)*, 2021, pp. 1–7.
- [19] R. Lent, "Implementing a cognitive routing method for high-rate delay tolerant networking," in *2023 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, 2023, pp. 1–6.
- [20] D. Raible, R. Dudukovich *et al.*, "Developing High Performance Space Networking Capabilities for the International Space Station and Beyond," NASA, Technical Memorandum, 2022. [Online]. Available: <https://ntrs.nasa.gov/api/citations/20220011407/downloads/TM-20220011407.pdf>
- [21] E. Birrane, K. Collins, and K. Scott, "The delay-tolerant networking engineering network - constructing a cross-agency supported internet-networking testbed," in *American Institute of Aeronautics and Astronautics*, 2012, pp. 1–11.
- [22] Apposite Technologies. Linktropy Network Delay Simulator. [Online]. Available: <https://www.apposite-tech.com/products/linktropy/>
- [23] —. Netropy Network Emulator. [Online]. Available: <https://www.apposite-tech.com/products/netropy-network-emulation/>
- [24] B. Hubert. Linux Advanced Routing Traffic Control. [Online]. Available: <https://lartc.org/>
- [25] Keysight Technologies. PROPSIM Channel Emulator Platforms. [Online]. Available: <https://www.keysight.com/us/en/products/channel-emulators/propsim-platforms.html>
- [26] J. Fraire, "Unity 3d visualization tool," <https://youtu.be/n2Z63EPaYvo>, IPNSIG, 2023, video published by IPNSIG.
- [27] U.S. Naval Research Laboratory. Common Open Research Emulator (CORE). [Online]. Available: <https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/CORE/demo1/>
- [28] U.S. Naval Research Laboratory (NRL). Extendable Mobile Ad-hoc Network Emulator (EMANE). [Online]. Available: <https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/EMANE/>
- [29] Keysight Technologies. EXata Network Modeling. [Online]. Available: <https://www.keysight.com/us/en/product/SN100EXBA/exata-network-modeling.html>
- [30] Wireshark Foundation. Wireshark. [Online]. Available: <https://www.wireshark.org/>
- [31] Ansys. Systems Tool Kit (STK). [Online]. Available: <https://www.ansys.com/products/missions/ansys-stk>
- [32] P. Apollonio, C. Caini, M. Giusti, and D. Lacamera, "Virtualbricks for dtn satellite communications research and education," in *6th International Conference on Personal Satellite Services (PSATS '14)*. Springer International Publishing, 07 2014, pp. 76–88.
- [33] E. J. Birrane, C. Caini, G. M. De Cola, F. Marchetti, L. Mazzuca, and L. Persampieri, "Opportunities and limits of moderate source routing in delay-/disruption-tolerant networking space networks," *International Journal of Satellite Communications and Networking*, vol. 40, no. 6, pp. 428–444, 12 2022. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1421>
- [34] M. Bertolazzi and C. Caini, "Mars to earth data downloading: A directory synchronization approach," *Future Internet*, vol. 11, no. 8, 8 2019. [Online]. Available: <https://www.mdpi.com/1999-5903/11/8/173>
- [35] H. Li, H. Zhou, H. Zhang, B. Feng, and W. Shi, "Emustack: An openstack-based dtn network emulation platform (extended version)," *Mobile Information Systems*, vol. 2016, 12 2016. [Online]. Available: <https://doi.org/10.1155/2016/6540207>
- [36] I. Komnios, I. Alexiadis, N. Bezirgiannidis, S. Diamantopoulos, S.-A. Lenas, G. Papastergiou, and V. Tsaoussidis, "Spice testbed: A dtn testbed for satellite and space communications," in *ICST Conference (TridentCom '14)*. Springer, 5 2014.
- [37] I. Komnios, S. Diamantopoulos, and V. Tsaoussidis, "Evaluation of dynamic dtn routing protocols in space environment," in *IEEE ISWCS'09*. IEEE, 09 2009, pp. 191–195. [Online]. Available: <https://doi.org/10.1109/IWSSC.2009.5286392>
- [38] G. Papastergiou, N. Bezirgiannidis, and V. Tsaoussidis, "On the performance of erasure coding over space dtns," in *10th International Conference on Wired / Wireless Internet Communications (WWIC '12)*. Berlin, Heidelberg: Springer, 6 2012, pp. 269–281.
- [39] N. Bezirgiannidis and V. Tsaoussidis, "Packet size and dtn transport service: Evaluation on a dtn testbed," in *International Conference on Ultra Modern Telecommunications and Workshops (ICUMT '10)*. IEEE, 10 2010, pp. 1198–1205.
- [40] Y. Li, P. Hui, D. Jin, and S. Chen, "Delay-tolerant network protocol testing and evaluation," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 258–266, 1 2015.
- [41] R. Lent and G. Velusamy, "Testing smart dtn routing using cloudlab," in *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2020, pp. 1–4.
- [42] L. C. Suzuki, V. G. Cerf, J. L. Torgerson, and T. S. Suzuki, "A novel federated computation approach for artificial intelligence applications in delay and disruption tolerant networks," in *2023 IEEE Cognitive Comms. for Aerospace Applications Workshop (CCAAW)*, 2023, pp. 1–8.
- [43] J. Miner. New Aerospace Communications Facility Enables Ambitious NASA Missions. [Online]. Available: <https://www.nasa.gov/general/new-aerospace-communications-facility-enables-ambitious-nasa-missions/>
- [44] J. Downey, A. Gannon, A. Smith, M. Koch, and R. Dudukovich, "Emulated spacecraft communication testbed for evaluating cognitive networking technology," in *CCAAW'23: IEEE Cognitive Comms. for Aerospace Applications Workshop*, 2023.
- [45] J. Nessel, J. Downey, and C. Niederhaus, "The Multiple Access Testbed for Research in Innovative Communications Systems (MATRICS)," in *ESA Antenna Workshop*, Noordwijk, Netherlands, October 2018.
- [46] A. Schlesinger, B. M. Willman, L. Pitts, S. R. Davidson, and W. A. Pohlchuck, "Delay/Disruption Tolerant Networking for the International Space Station (ISS)," in *2017 IEEE Aerospace Conf.*, 2017, pp. 1–14.

- [47] The National Aeronautics and Space Administration. Technology Readiness Levels. [Online]. Available: <https://www.nasa.gov/directorates/somd/space-communications-navigation-program/technology-readiness-levels/>
- [48] R. Dudukovich, G. Clark, and C. Papachristou, "Evaluation of classifier complexity for delay tolerant network routing," in *2019 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, 2019, pp. 1–7.
- [49] M. Blanchet, L. Torgerson, and Y. Qu. Contact Plan Yang Model for Time-Variant Routing of the Bundle Protocol. [Online]. Available: <https://datatracker.ietf.org/doc/html/draft-blanchet-tvr-contactplan-01>