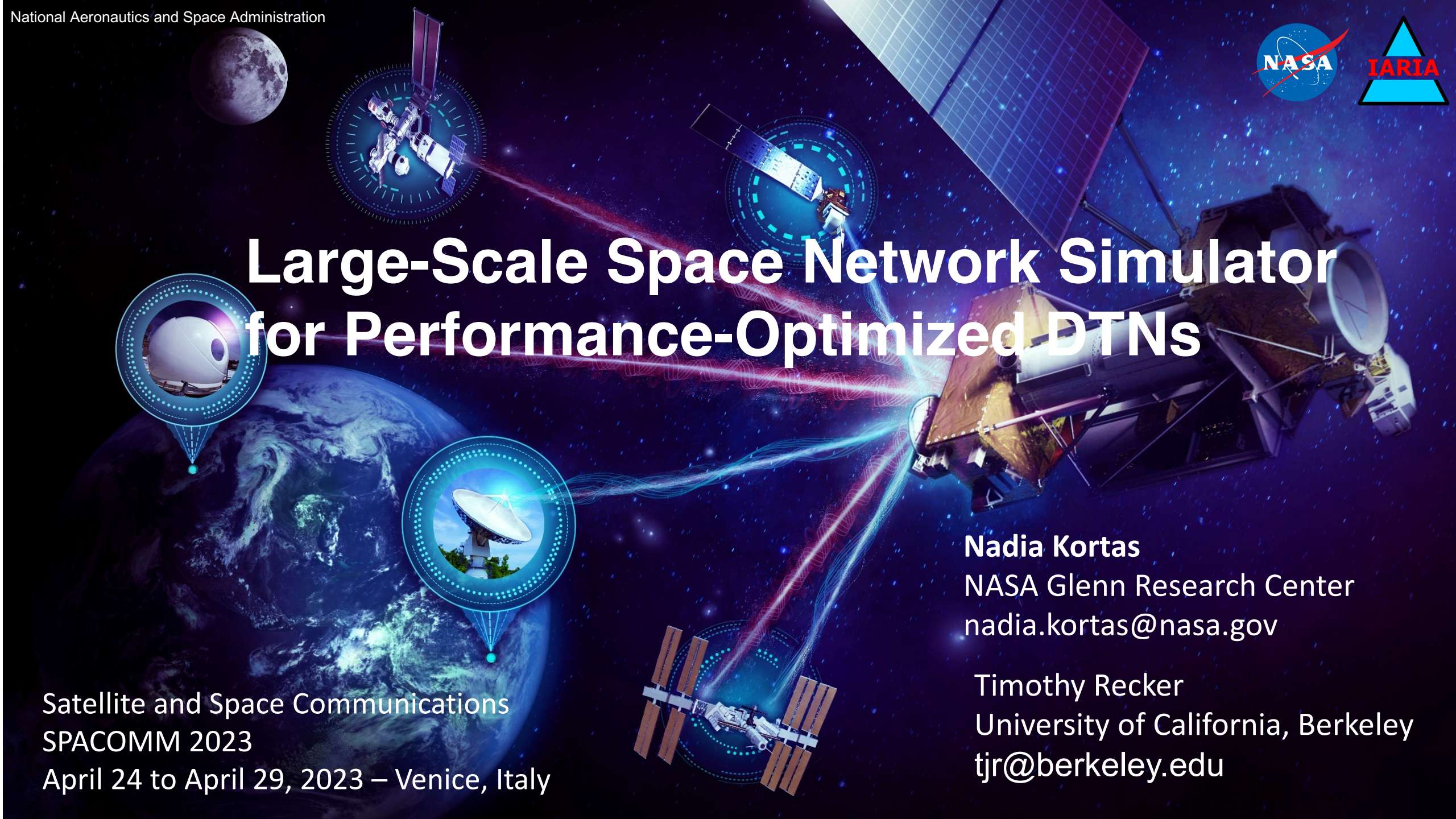
National Aeronautics and Space Administration



Large-Scale Space Network Simulator for Performance-Optimized DTNs

Nadia Kortas
NASA Glenn Research Center
nadia.kortas@nasa.gov

Timothy Recker
University of California, Berkeley
tjr@berkeley.edu

Satellite and Space Communications
SPACOMM 2023
April 24 to April 29, 2023 – Venice, Italy

Meet the Authors

Nadia Kortas

Nadia Kortas is a research computer engineer working on the development of software to support embedded space communication systems and other spaceflight and science projects at NASA. She's currently leading the design and implementation of High-rate Delay Tolerant Networking software routing and security. She holds an M.S. in Engineering and Computer Science from Mines ParisTech, France.

Timothy Recker

Timothy Recker is a NASA Intern and fourth-year undergraduate studying electrical engineering and computer science at the University of California, Berkeley. He graduates May 2023 and is eager to apply what he's learned to work on embedded systems, communication networks, and digital circuits.

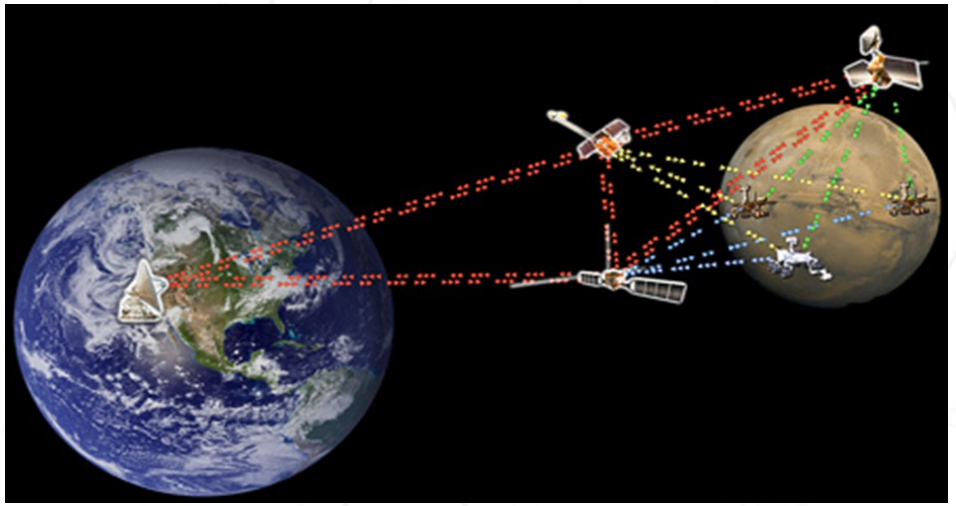
Introduction

Terrestrial internet:

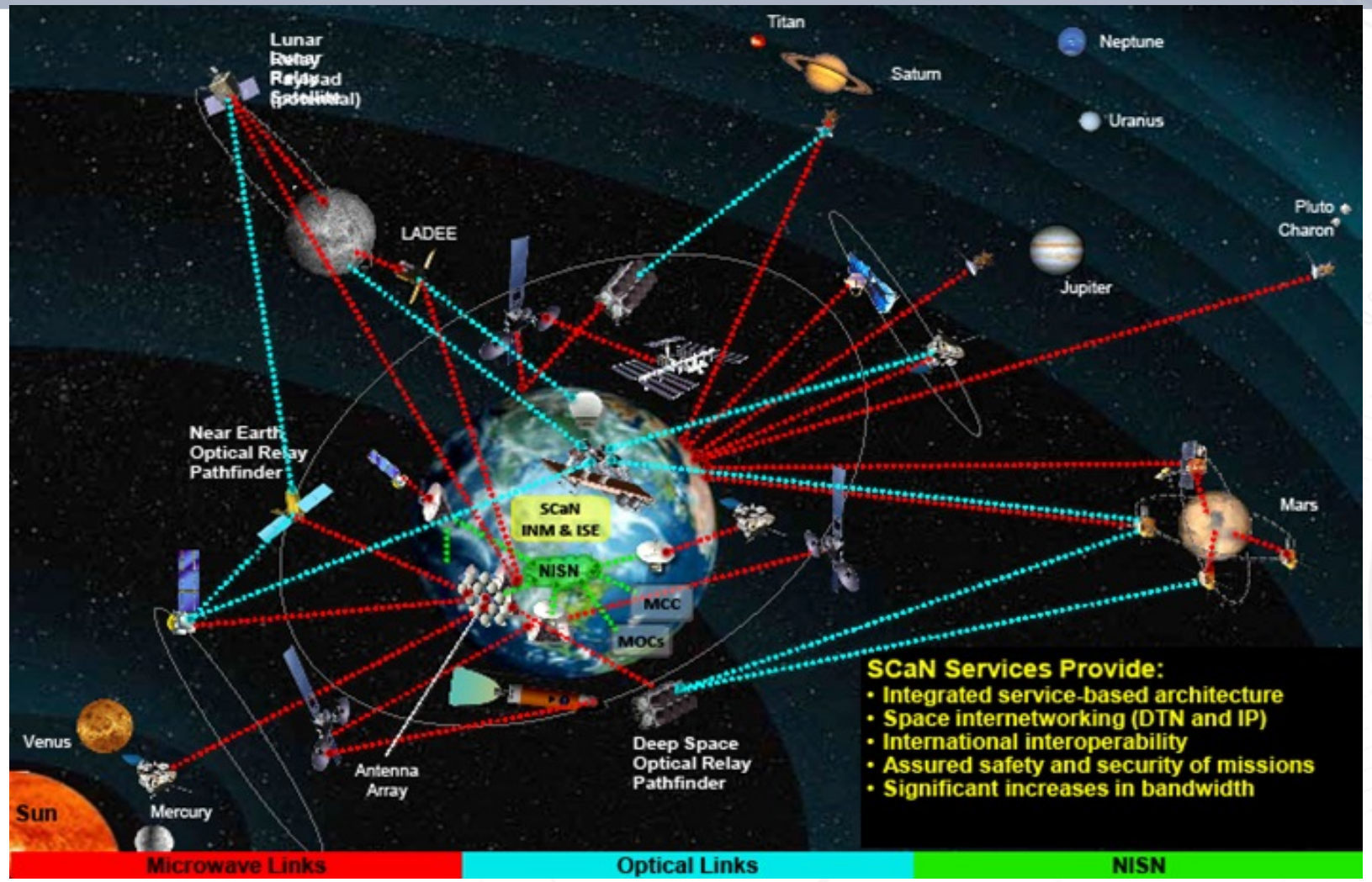
- Continuous end-to-end connectivity
- Very brief round trip times (RTT): 100ms to 300ms
- Data links are symmetric

Solar system internet:

- Disrupted networks without end-to-end connectivity even in near-Earth space (constant orbital movements)
- Very long RTT: Moon 2560ms; Mars from 6 min to 44.6 min ⟹ negating ACK -based protocols (TCP/IP)
- High rates of data loss
- Link data rates are often asymmetric

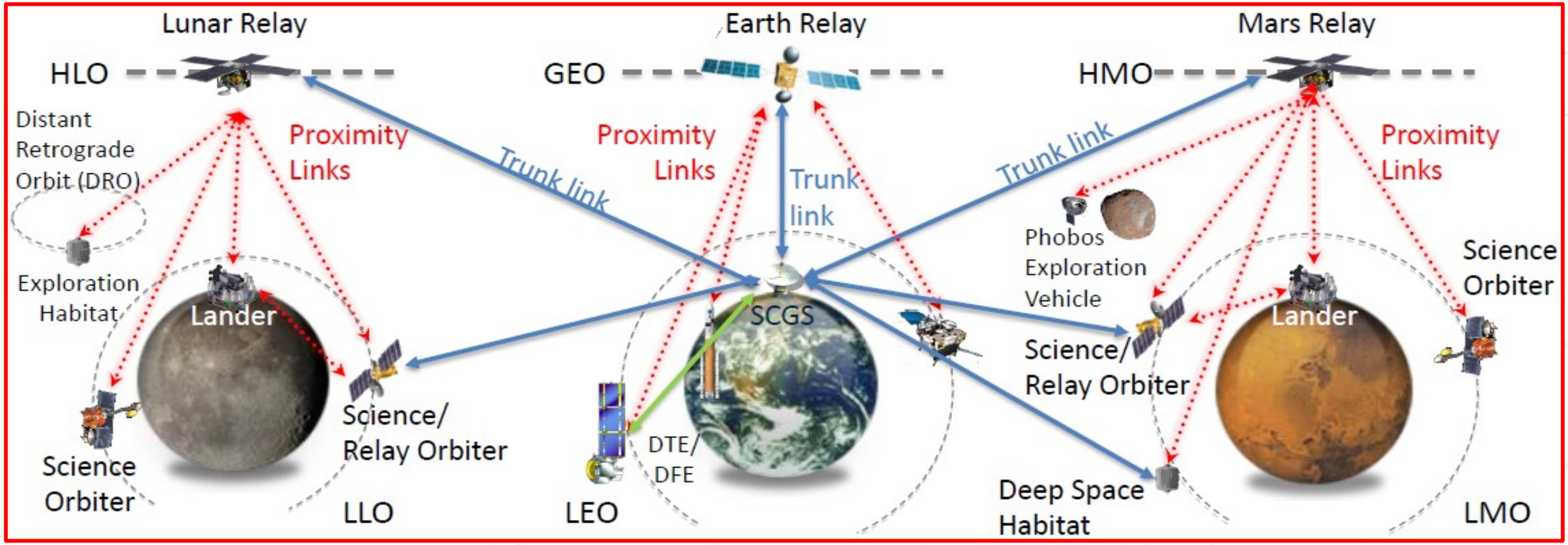


Point to Point Space-Terrestrial Links



Network Topology Transformation

Topology Transformation: Planetary Networks



Delay/Disruption Tolerant Networking (DTN)

DTN is NASA's solution for automated and reliable communication for space missions:

- ✓ Unites the formerly disjointed collection of assets and links to form a cohesive system of systems
- ✓ Buffers data until a transmit opportunity arises, where an end-to-end path may not be available
- ✓ Uses the bundle protocol, which forms a store-and-forward overlay network (RFC 4838 and RFC 5050)

Store,
Carry,
Forward…

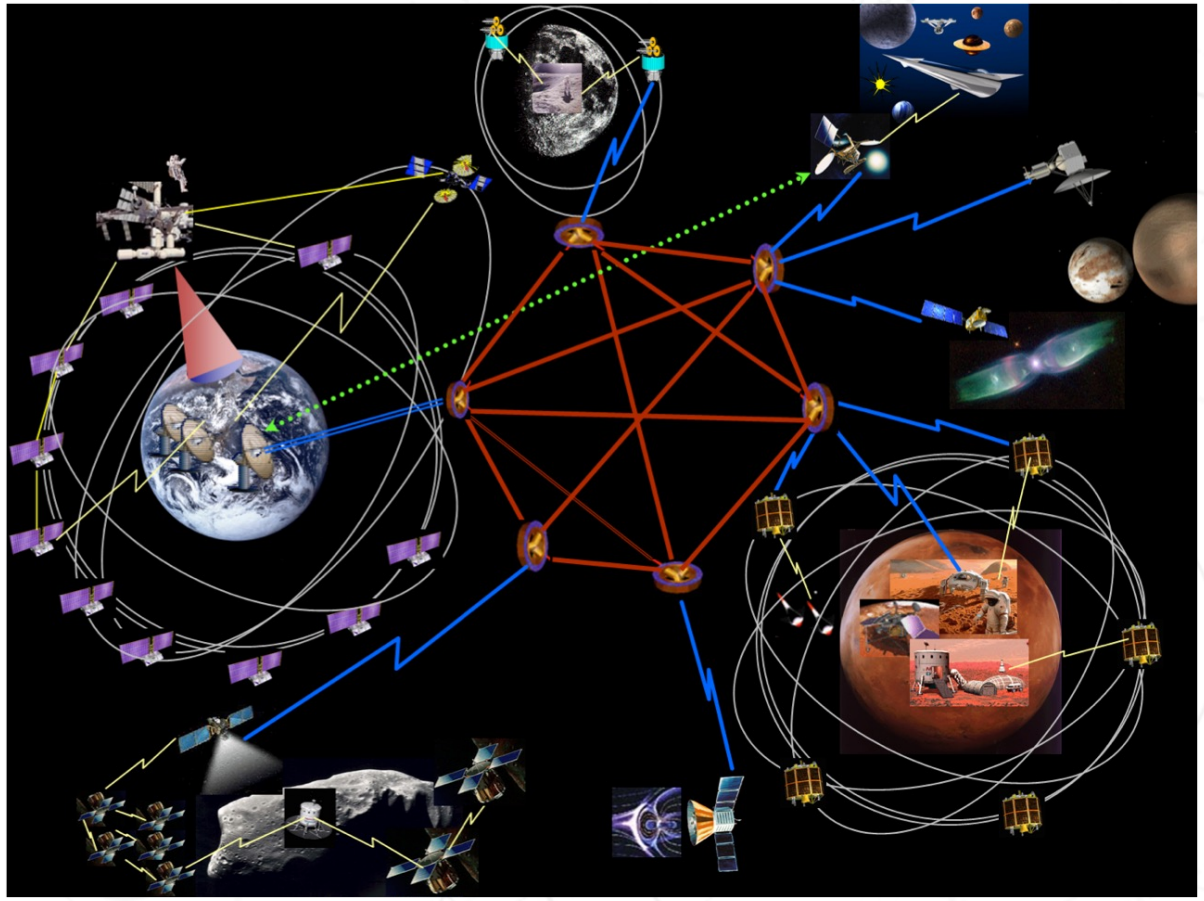
High-Rate Delay Tolerant Networking (HDTN)

- HDTN is a performance-optimized implementation of DTN by NASA Glenn Research Center that can provide high data rates on the order of multiple Gbps
- HDTN targets applications involving high-rate RF and optical links to help increase the amount of science data returned on future space missions

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field

Large Scale of Space Communication Networks

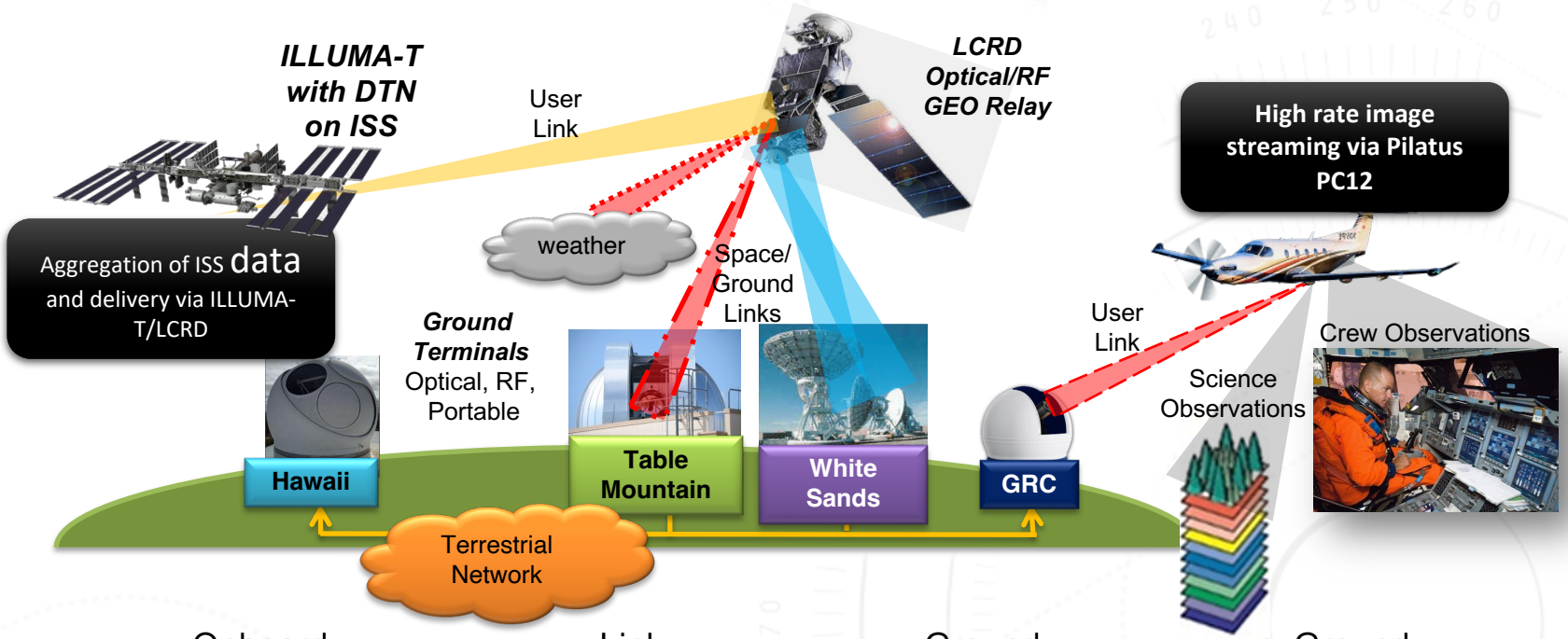
- The scale of space communication networks continues to increase
- Routing in these networks is more challenging due to large topologies that evolve over time
- HDTN has been developed with the goal of scaling to large, heterogeneous, interplanetary networks while maximizing performance

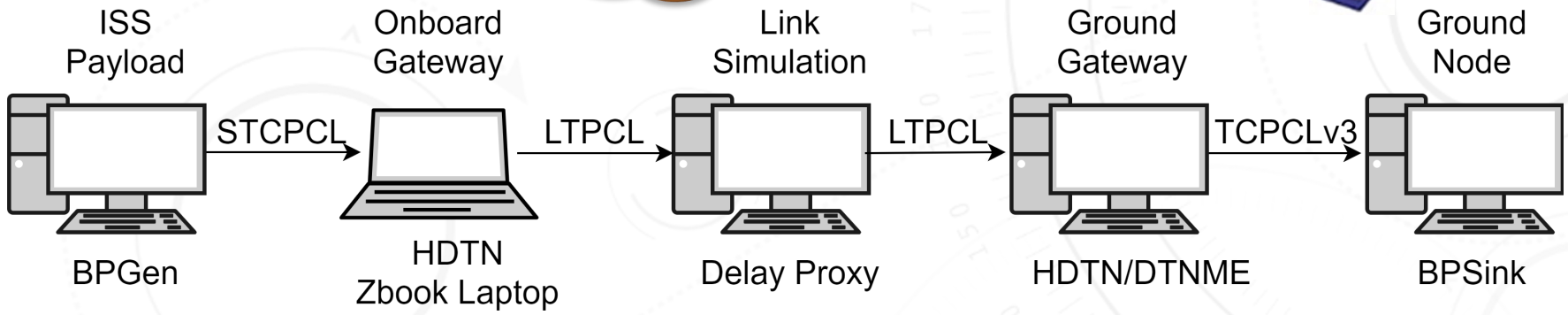


Evaluating HDTN in the Laboratory

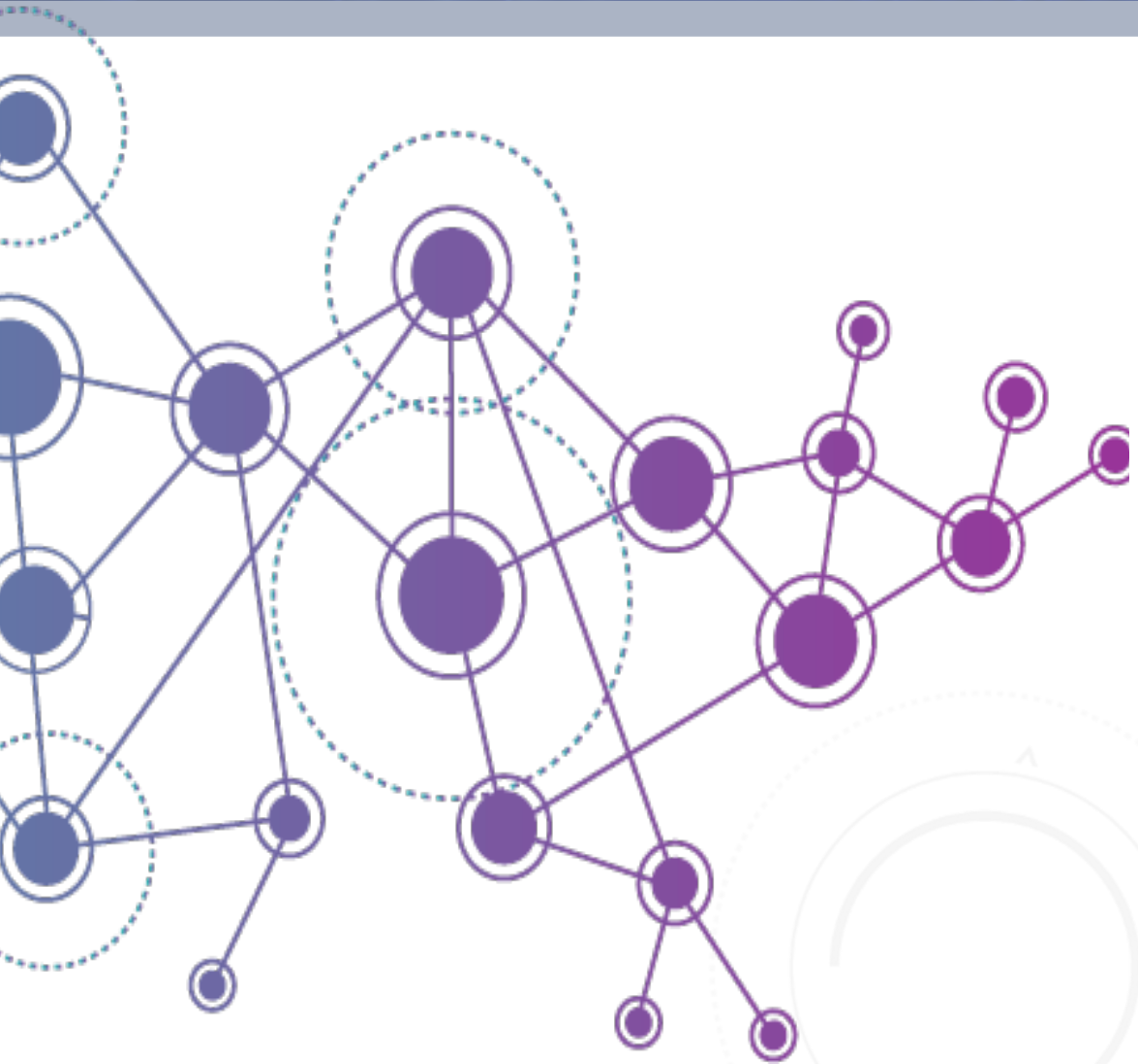
Laboratory Emulated Testbed

Enabling Networked Optical Communications Rates





Problem Statement



Network of HDTN nodes testing methods

- Hardware lab tests, emulation on virtual machines
- Scale: up to 10 nodes
- Timescale: prolonged experimental time

Laboratory tests: Slow, difficult to configure/set up and did not scale to large numbers of nodes or long time periods

➔ A simulation tool was needed to enable HDTN implementation and testing at accelerated speeds, which is key for routing in large-scale space networks

Goals

01

Speed up the setting up and configuring of tests

02

Speed up the running of tests

03

Scale to large contact plans and longer time periods

Agenda

HDTN architecture

HDTN simulator design

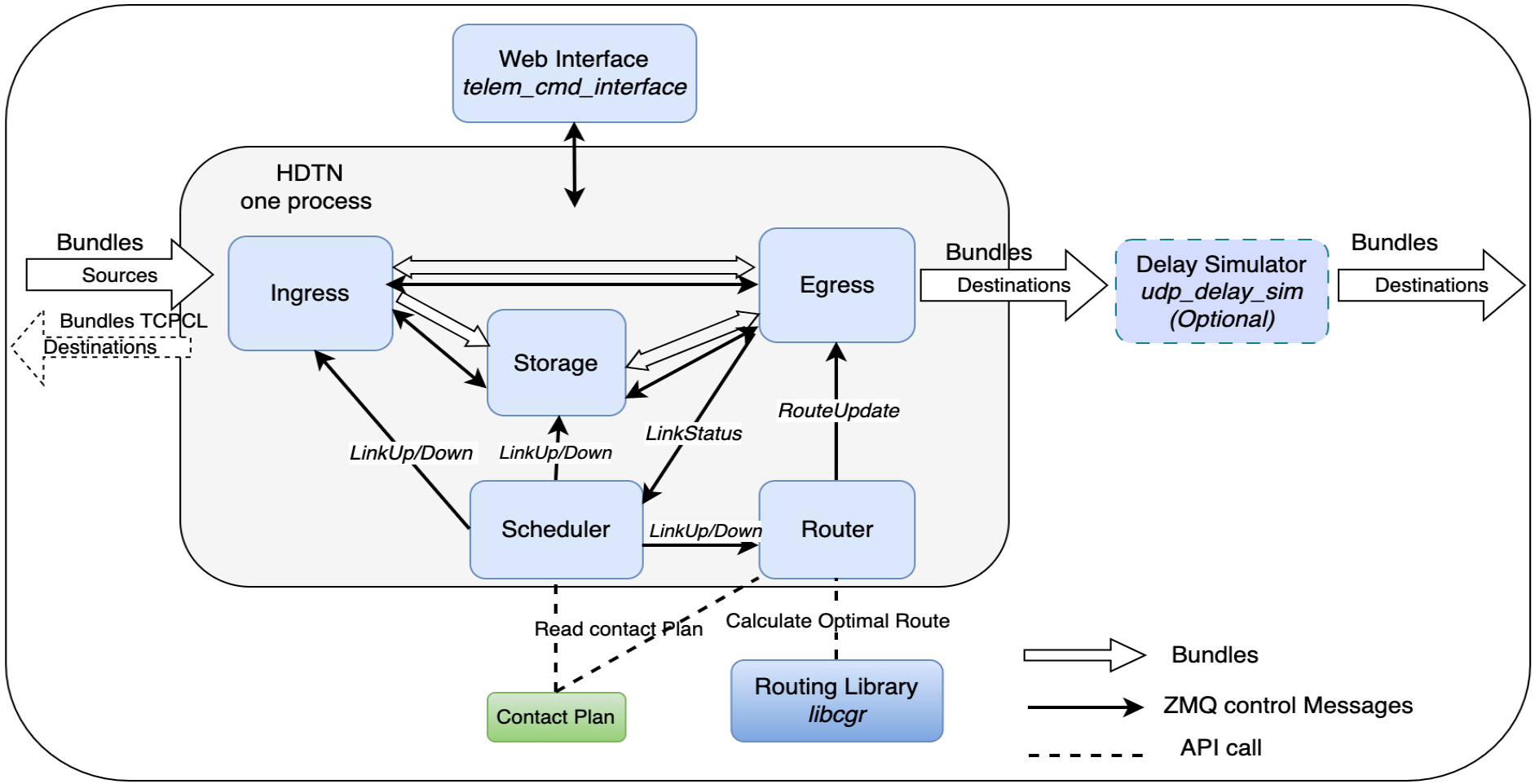
Simulator vs. Runscript results

CGR vs. CMR results

HDTN vs. ION results

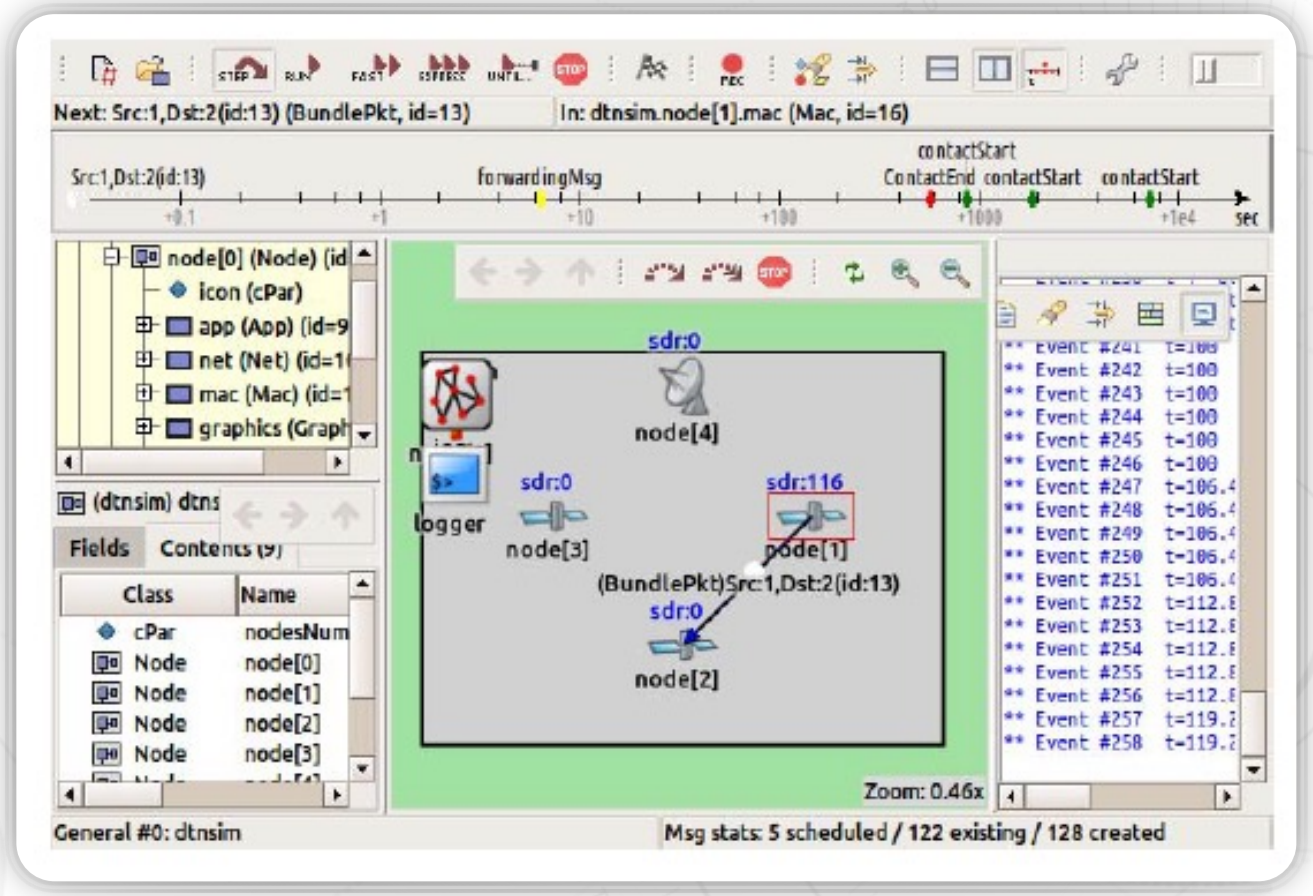
Scaling to large networks

HDTN: Software Architecture

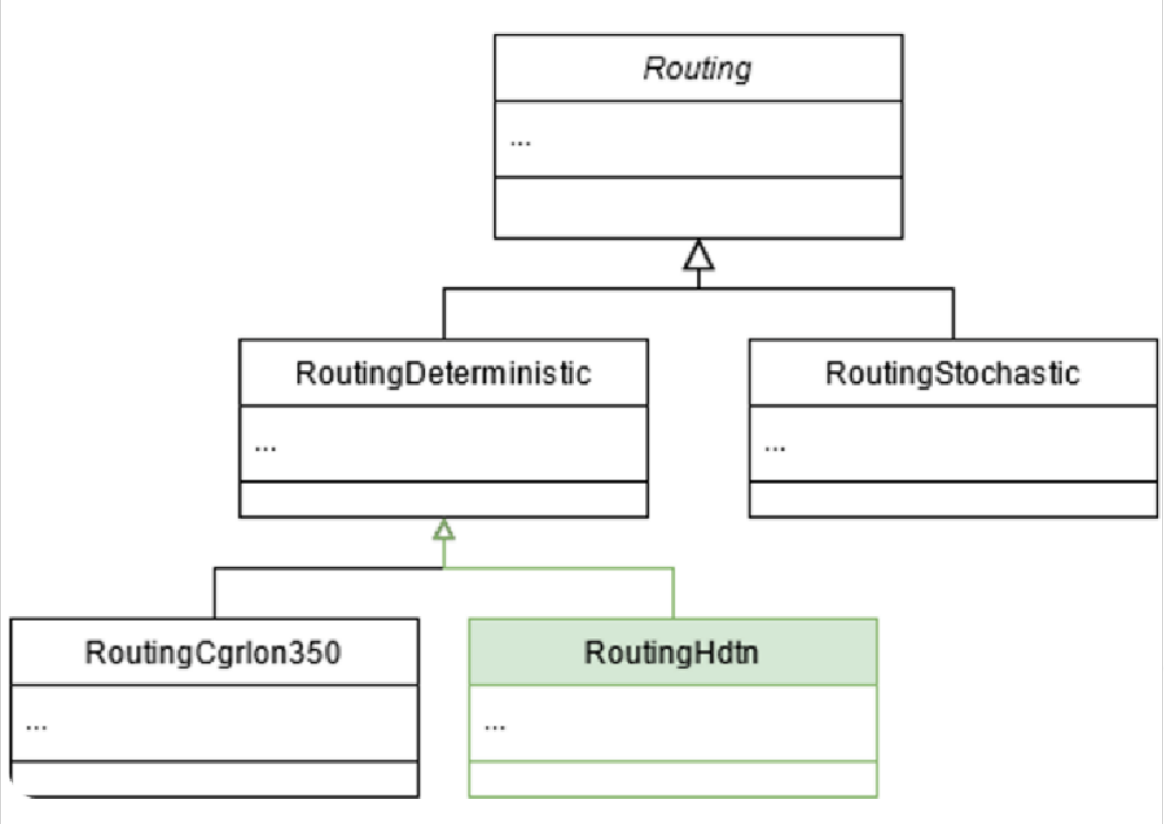


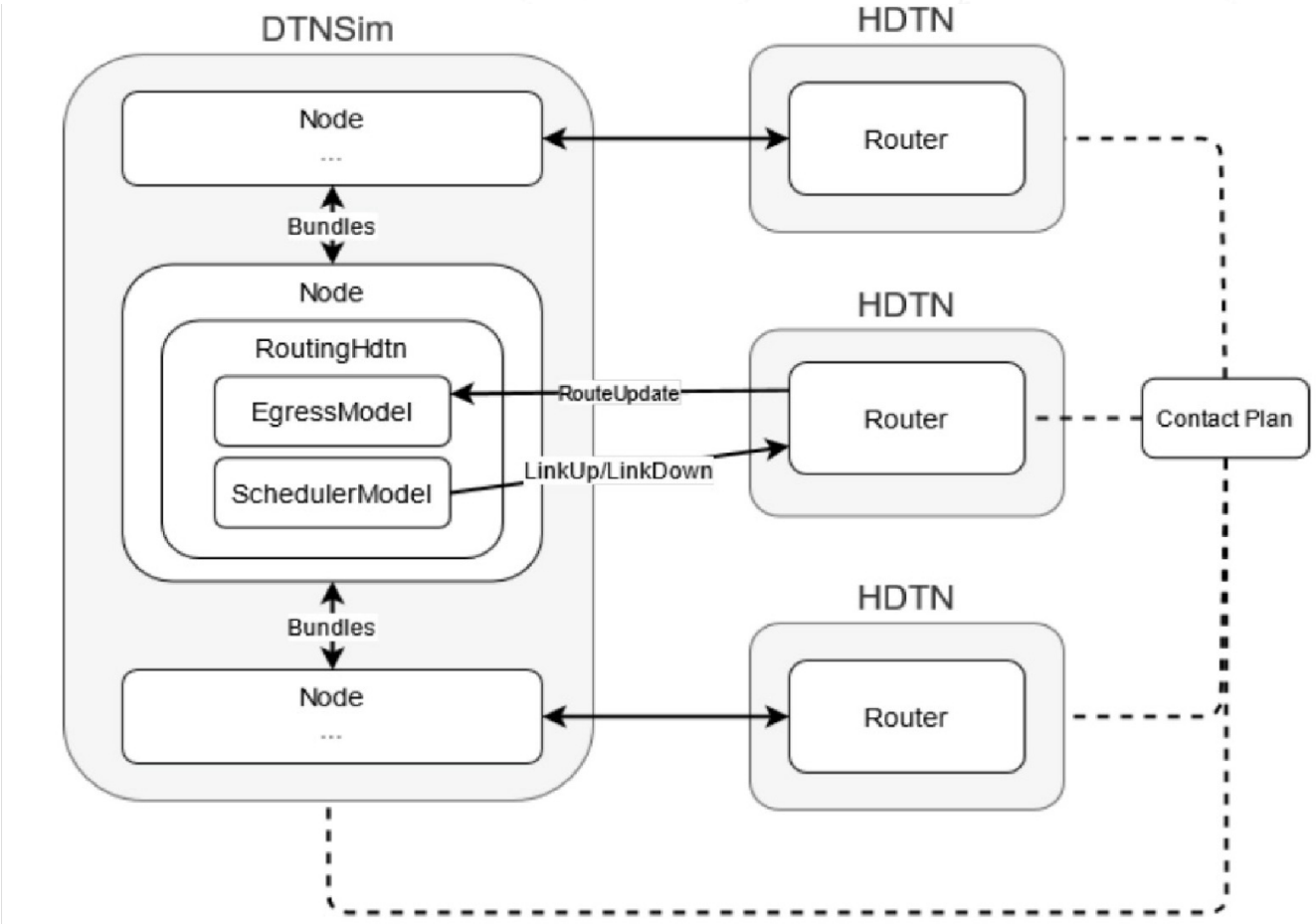
DTNSim

- A DTN simulator built on OMNeT++ framework
- Visualization, metrics collection, network flow diagram generation, and development environment
- Numerous routing approaches supported, including ION routing

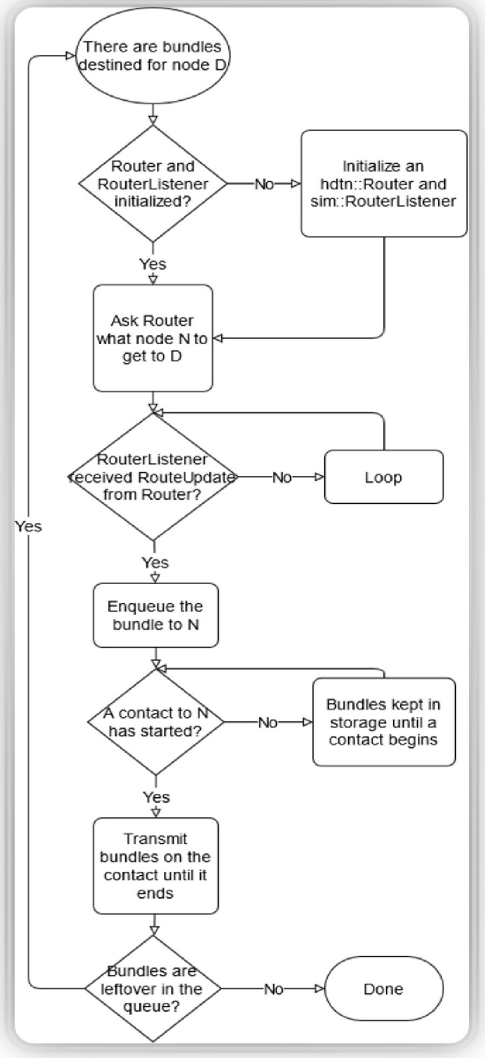


(H)DtnSim: Software Architecture



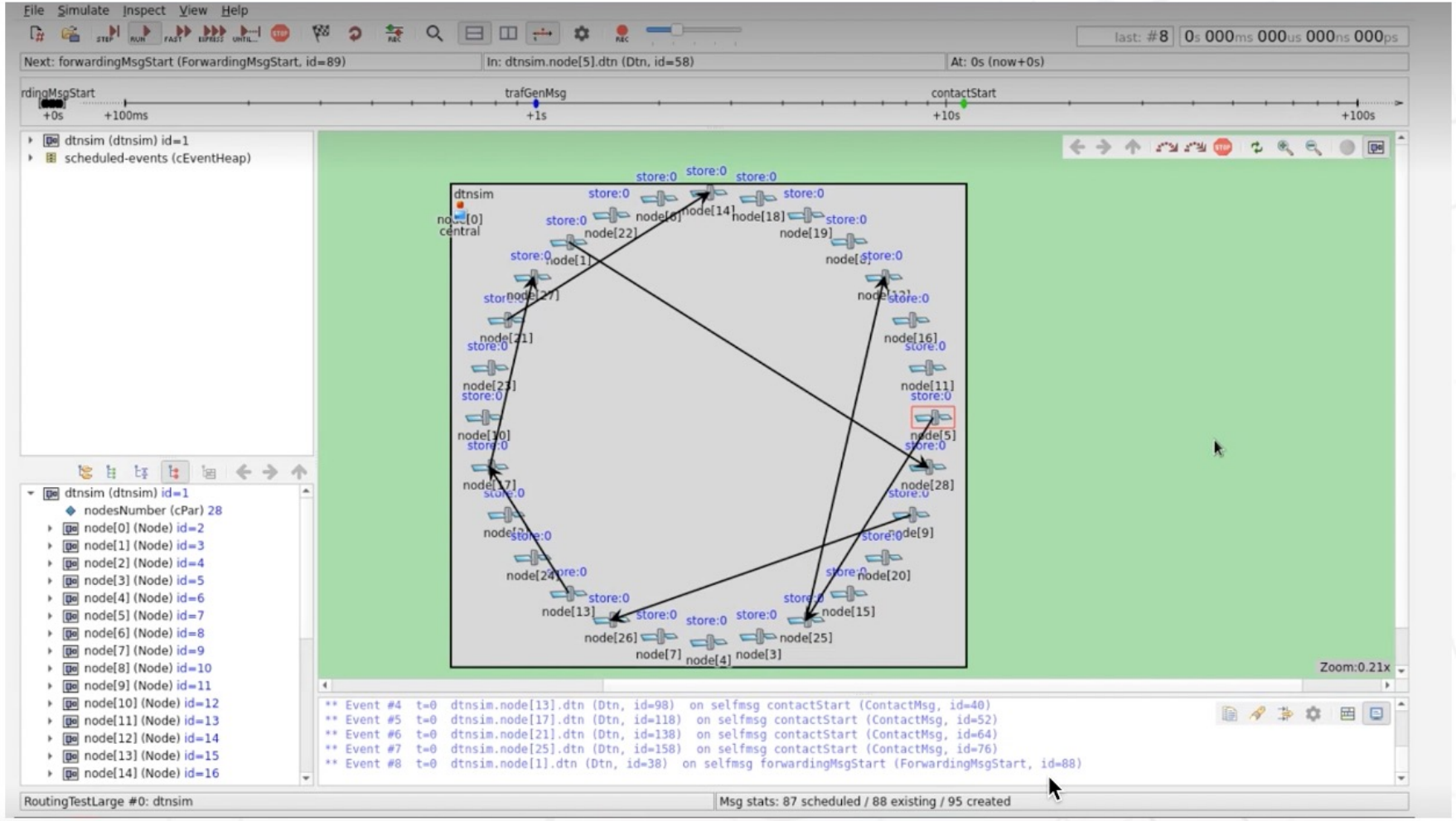


(H)DtnSim: Flow Diagram



- Questions a simulator can answer
  - What is the delivery rate, throughput, average delay, average hop count, for a network of N nodes using a topology T and running workload W
- Questions a simulator implementor must answer
  - How do I get a bundle from source node S to destination node D?
  - What would HDTN do? What would ION do?
  - How do we translate between ION, HDTN, and DtnSim models of concepts like storage, routing, forwarding, etc.?

(H)DtnSim: User Interface



Experimental Results: Simulation vs. Runscript

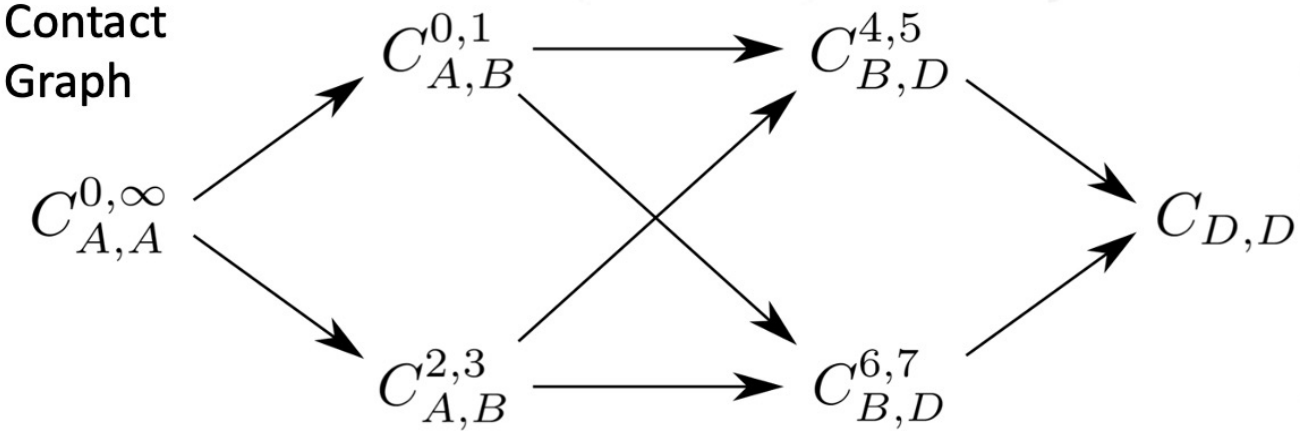
	Simulation	Runscript
Routes Found	$10 \rightarrow 2,\ 10 \rightarrow 1$	$10 \rightarrow 2,\ 10 \rightarrow 1$
Actual Bundles delivered	3800	3840 ± 6
Config lines (SLOC)	13	158
Run time (s)	1	73 ± 3
Discrete events	34558	NA

- Simple runscript.sh test copied from HDTN main repository to simulator
- Equivalent (approximately) results with large reduction in run time and configuration

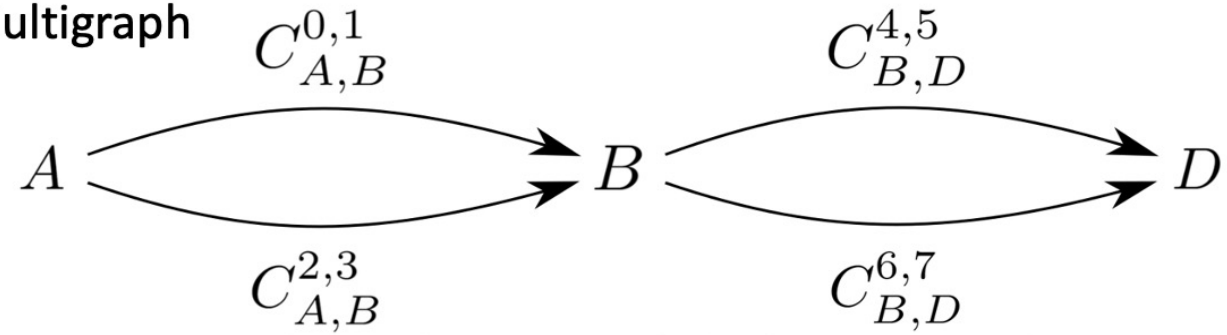
Experimental Results: CGR vs. CMR

- HDTN routing algorithms: Contact Graph (CGR) and Contact Multigraph Routing (CMR)
- Simulator works with both routing implementations and multiple HDTN versions
- Internal implementation is abstracted by messaging interface and has no effect on simulator functionality

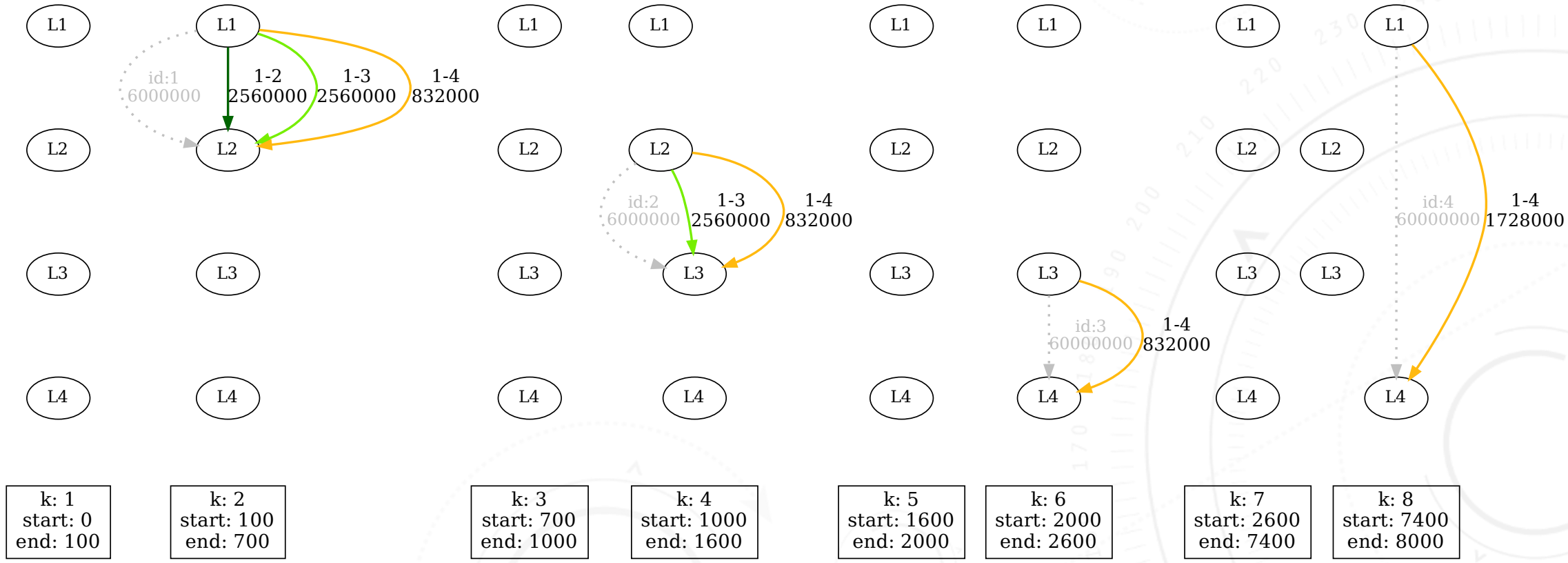
Contact Graph



Contact Multigraph

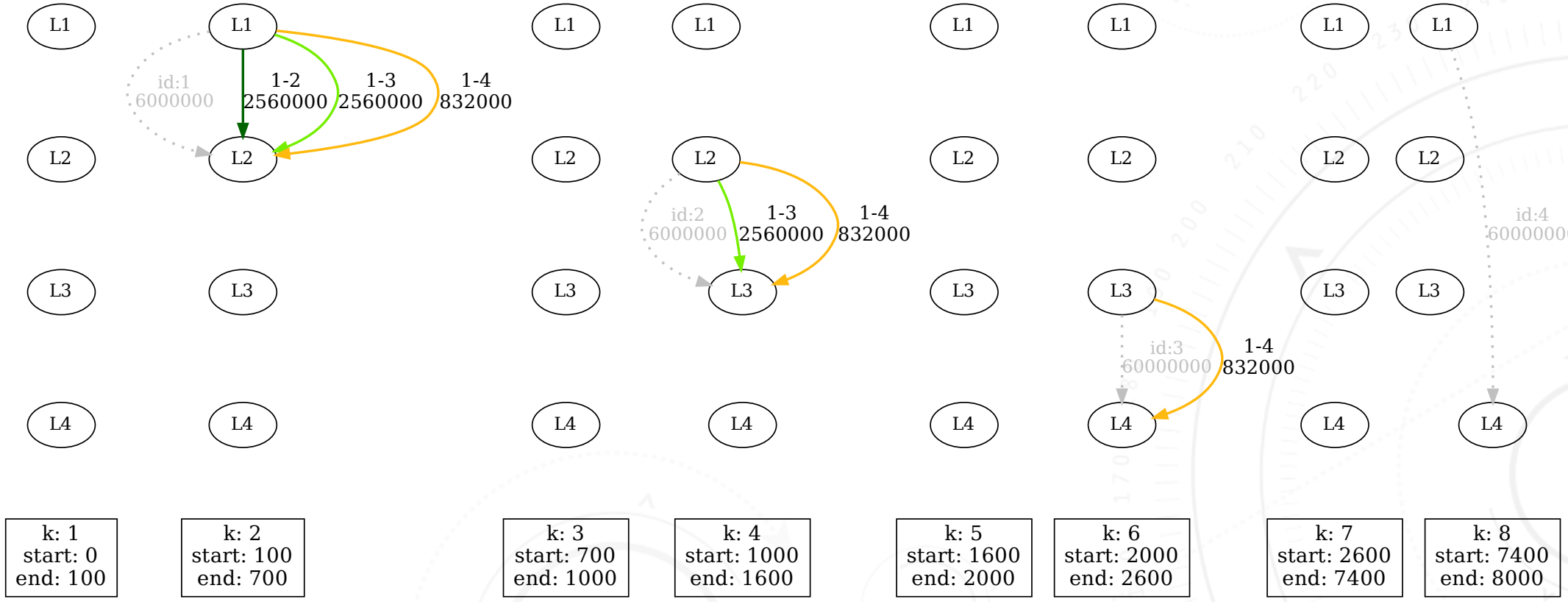


Experimental Results: HDTN vs. ION



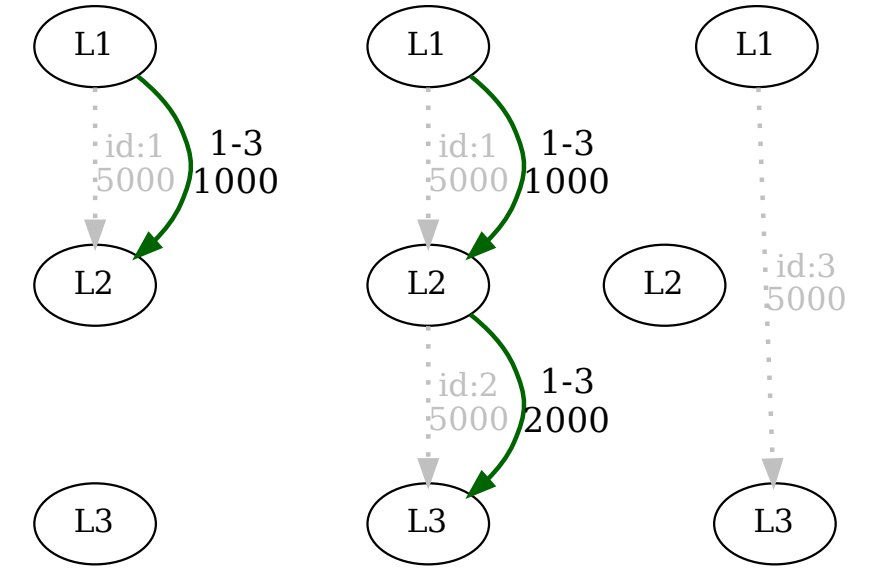
Dotted Lines = Contacts
Coloured Lines = Traffic Flows

Experimental Results: HDTN vs. ION



Dotted Lines = Contacts
Coloured Lines = Traffic Flows

Experimental Results: HDTN vs. ION

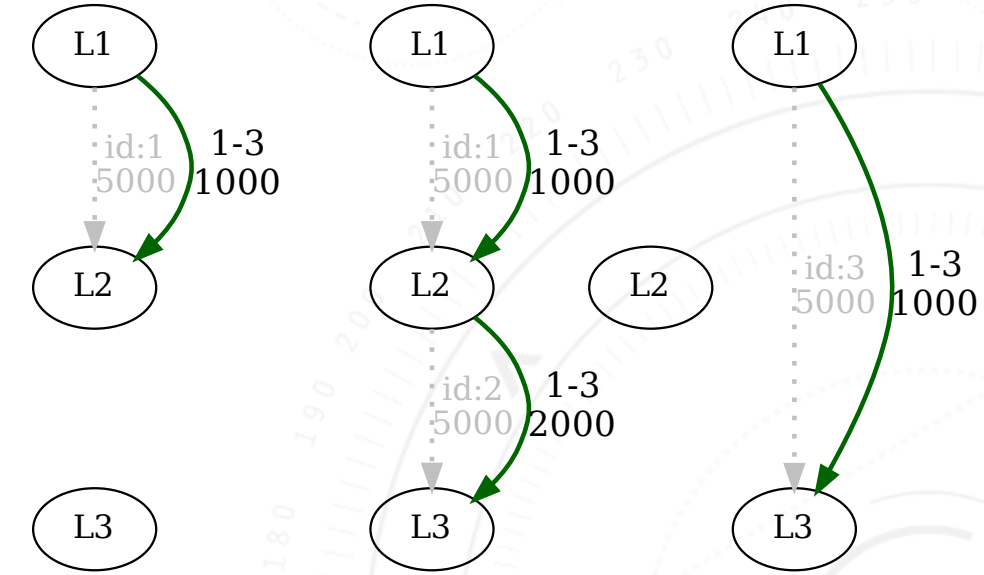


k: 1
start: 0
end: 5

k: 2
start: 5
end: 10

k: 3
start: 10
end: 15

Dotted Lines = Contacts
Coloured Lines = Traffic Flows



k: 1
start: 0
end: 5

k: 2
start: 5
end: 10

k: 3
start: 10
end: 15

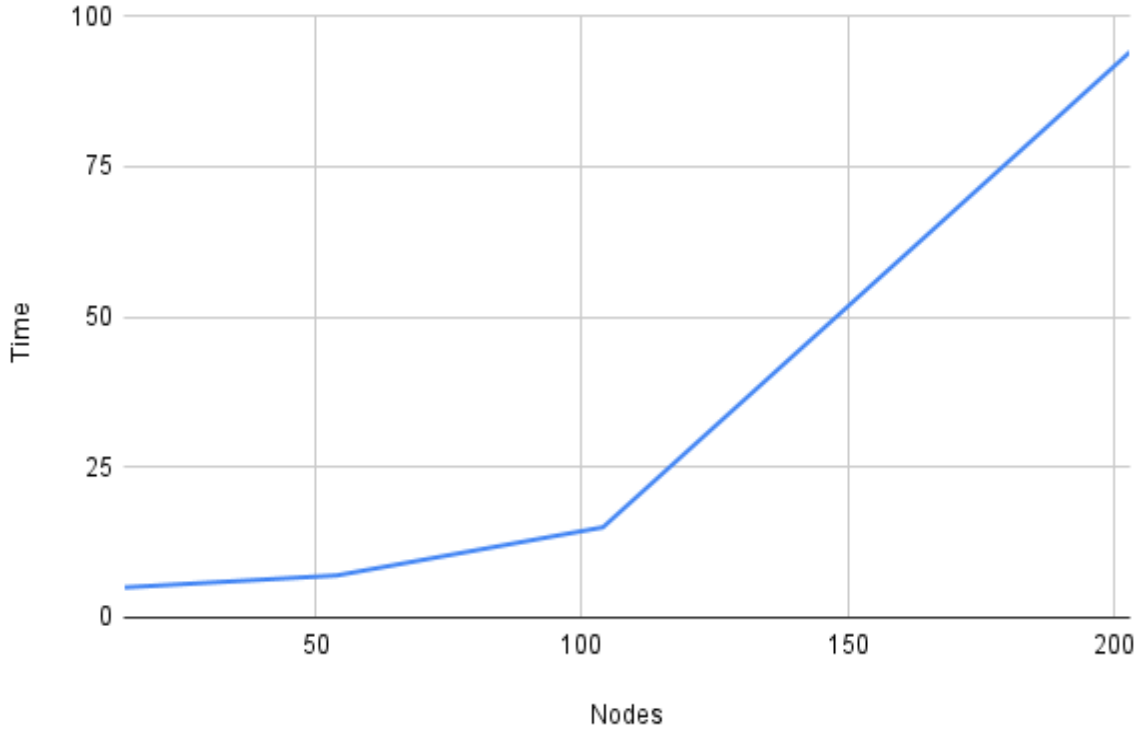
Dotted Lines = Contacts
Coloured Lines = Traffic Flows

Experimental Results: Large Contact Plans

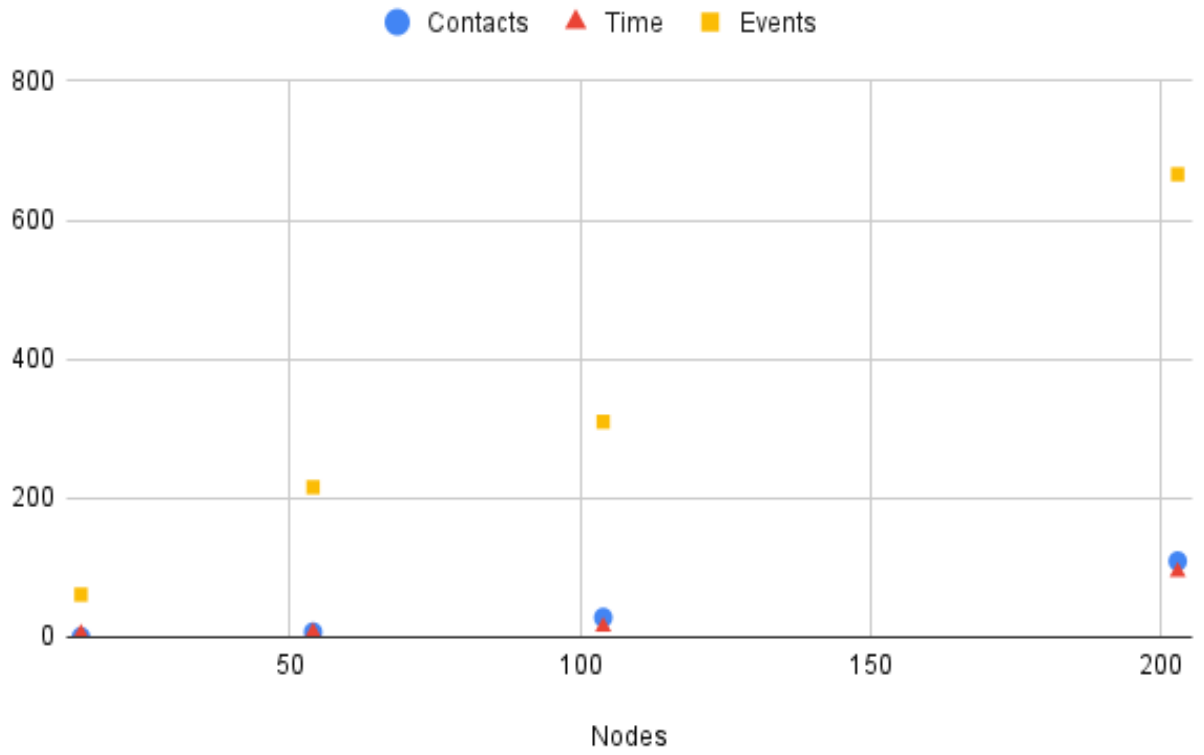
Nodes	14	54	104	203
Contacts	368	7186	28162	109329
Time (s)	5	7	15	94
Discrete events	611460	2157761	3098460	6658661

- Network topology based on data from Starlink satellite orbits
- Four ground station nodes in each run, 10–159 satellites orbiting
- For 20 seconds, ground stations send 1,907 bundles per second (65,535 bytes) via satellites
- Simulated time: 24 hours

Experimental Results: Large Contact Plans



Time (s) vs. number of nodes



Contacts (in 1,000s), time (s), and number of events (in 10,000s) plotted on one graph to show relationship between curve slopes

Conclusion

(H)DtnSim is a successful solution to the problems of testing networks of HDTN nodes:

- ✓Enabled HDTN routing testing at accelerated speeds
- ✓Scaled to large space networks
- ✓Worked with different HDTN versions and routing implementations
- ✓Provided user-friendly interfaces for the visualization and control of simulation scenarios and the analysis and plotting of metrics ⇒ produced enhancements to the HDTN Router's handling of time, improving bundles delivery rate
- ✓(H)DtnSim is publicly available:

  https://bitbucket.org/lcd-unc-ar/dtnsim/src/support-hdtn/

Questions?

References

“Delay-Tolerant Networking Architecture,” RFC 4838 https://datatracker.ietf.org/doc/rfc4838/

“Bundle Protocol Specification,” RFC 5050, https://www.rfc-editor.org/rfc/rfc5050/

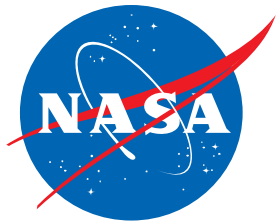
“New Horizons for a Practical and Performance-Optimized Solar System Internet,” https://doi.org/10.1109/AERO53065.2022.9843598/

HDTN Website with code links and publications: https://www1.grc.nasa.gov/space/scan/acs/tech-studies/dtn/

HDTN Github: https://github.com/nasa/HDTN/

DTNSim: https://bitbucket.org/lcd-unc-ar/dtnsim/src/support-hdtn/

HDTN Example Simulation in DTNSim: https://bitbucket.org/lcd-uncar/dtnsim/src/master/


NASA