Applying the Cognitive Space Gateway to Swarm Topologies

Ricardo Lent

University of Houston

Rachel Dudukovich, Adam Gannon, Bob Short

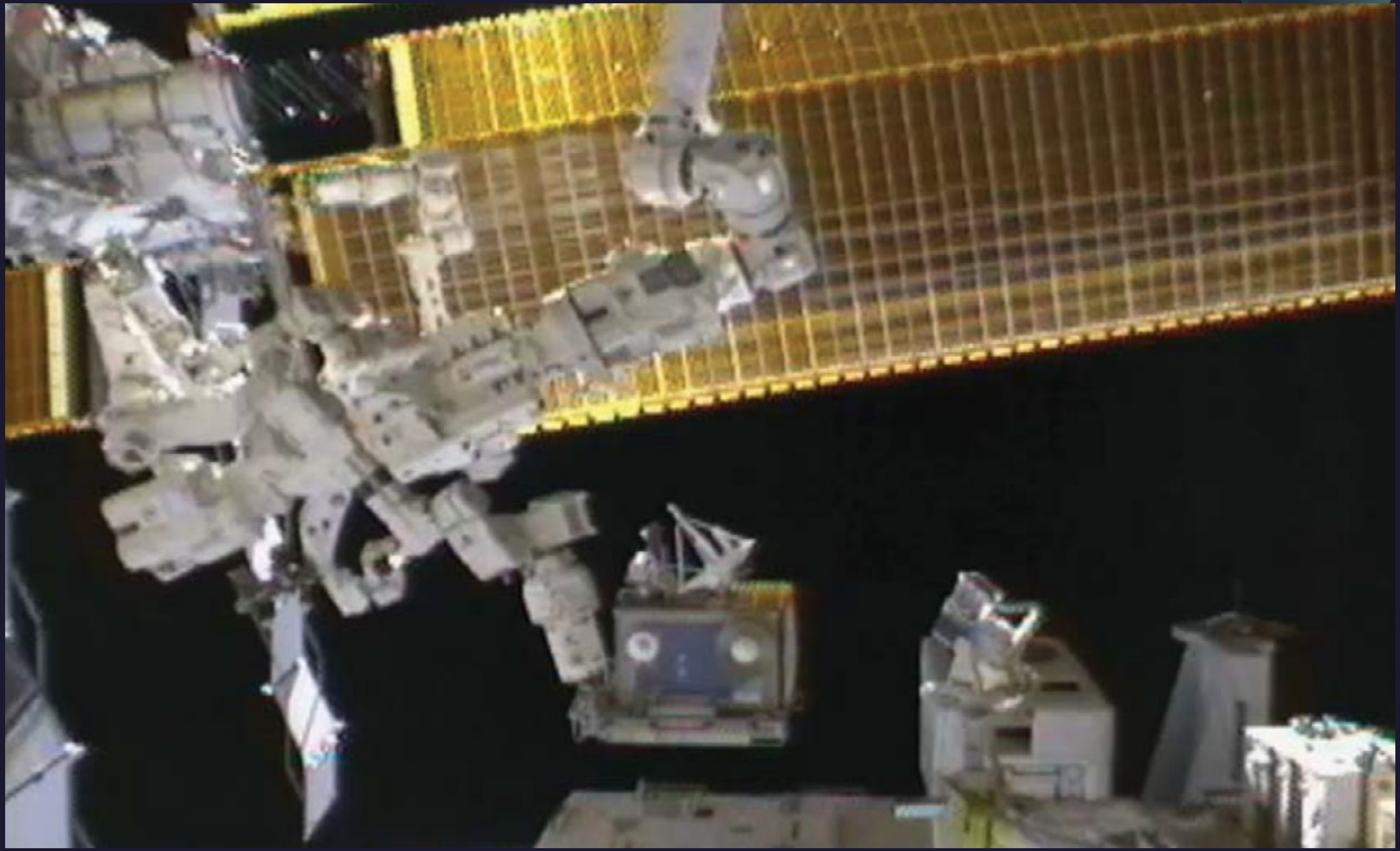
NASA Glenn Research Center

Cognitive Communications for Aerospace Applications Workshop

June 2021

Motivation

- The Cognitive Communications project seeks to mature the TRL of multiple network optimization techniques
- The Cognitive Space Gateway (CSG) builds upon the Cognitive Network Controller validated on SCaN testbed flight experiment

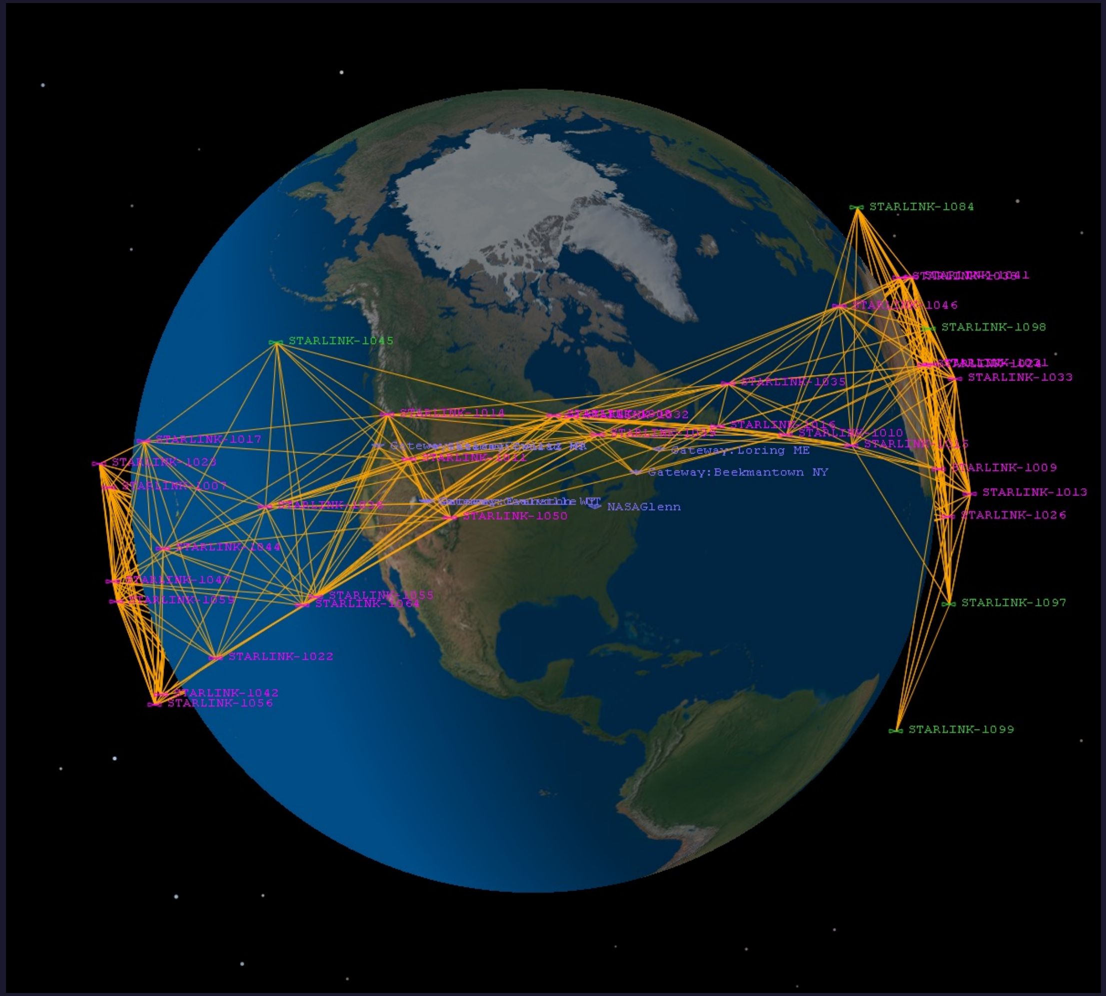


NASA Digital Transformation

- Ongoing effort seeks to embrace the power of digital technologies:
    - Cloud computing (including micro services, containers, etc.)
    - Artificial intelligence/machine learning
    - Collaborative tools
    - Many others
- We have taken this approach to collaborate with academic partners to advance cognitive networking technologies
- Develop multi-hop cloud testbed for cognitive network experiments
- Enable knowledge transfer in a virtual environment

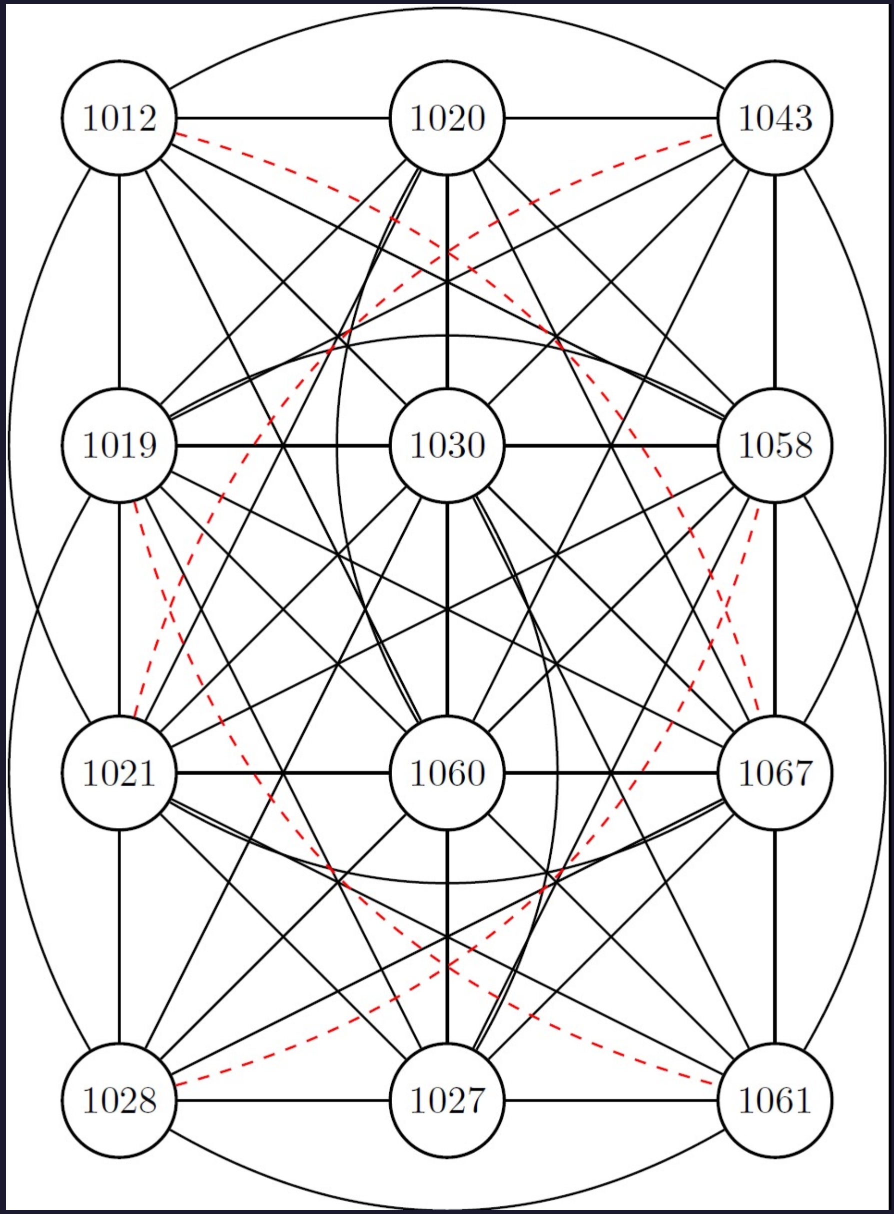
Network Modeling

- We drew inspiration from real missions, projects, and programs including Starling, HelioSwarm, **Starlink**, LunaNet, and TechEdSat
- From these, we estimated reasonable scenario parameters such as:
  - Number of nodes
  - Network topology
  - Contact schedules
  - Data rates
  - Data volume
  - Node storage and processing capabilities
- We constructed simulations using Satellite Orbital Analysis Program (SOAP)



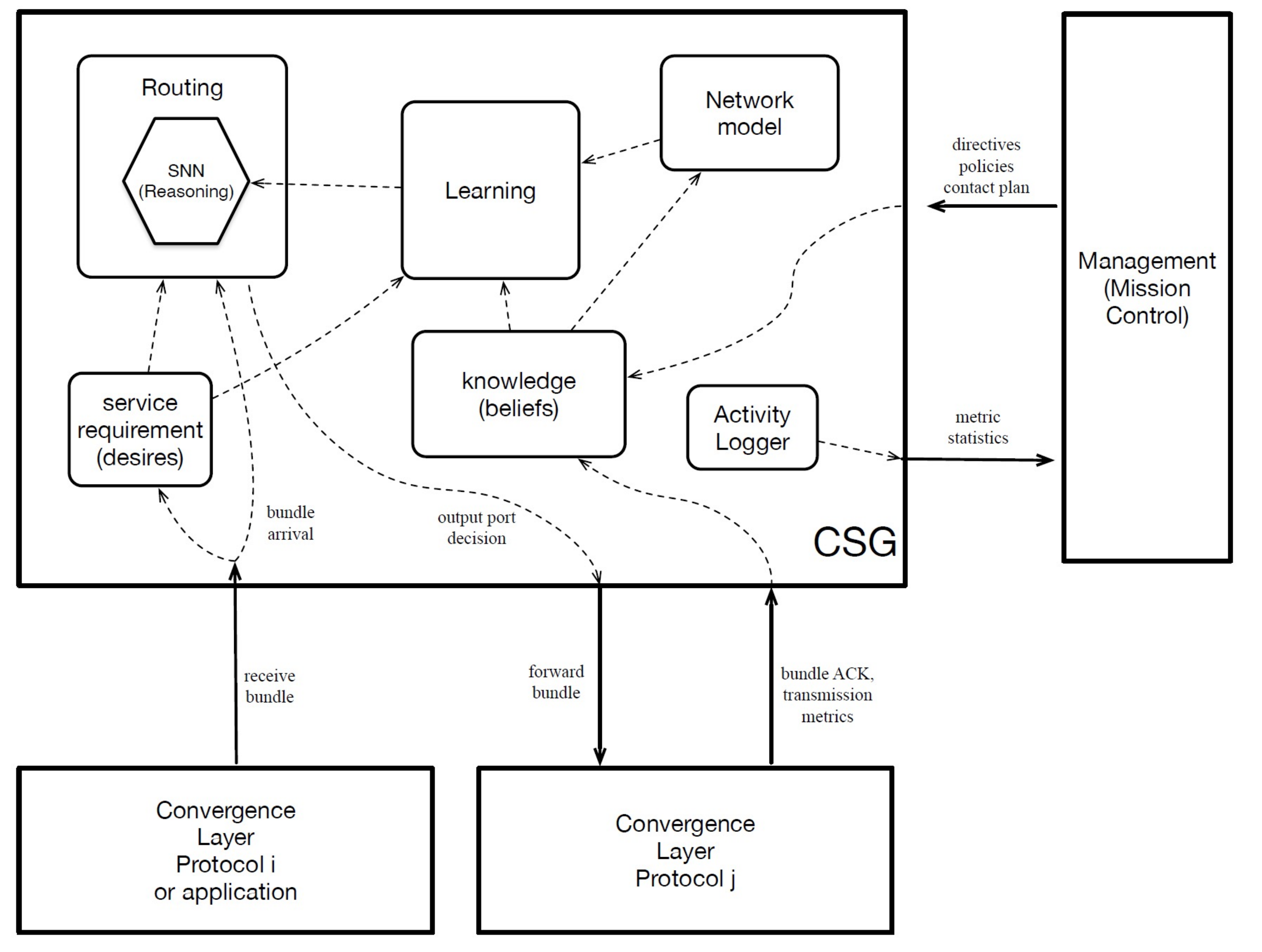
Experiment Topology

- We selected 12 nodes from the Starlink model in SOAP
- The CSG takes contact start time, stop time and average distance between nodes as inputs
- Propagation delay of the links calculated using the average distance between each pair of satellites within the satellite orbit simulation
- Node 1020 is the bundle source and node 1061 is the bundle sink
- In the selected scenario, there exists large path redundancy between any two nodes
    - 4 of the 57 network links are affected by link disruptions according to the network model



Cognitive Space Gateway (CSG)

- The CSG uses situated artificial intelligence to determine the optimal outbound link for data bundles
- Makes autonomous (near) optimal decisions on a per-bundle basis
- The Cognitive Network Controller (CNC) consists of a spiking neural network (SNN) that works as the learning and decision-making element for the CSG
- Reward shaping determines instantaneous rewards for reinforcement learning that continually adapt the CNC



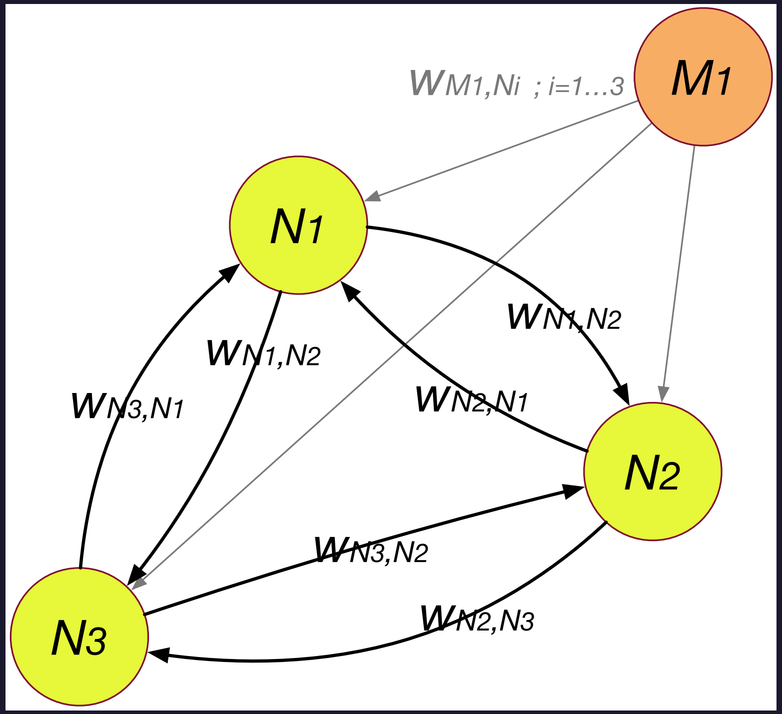
Cognitive Network Controller (CNC)

- SNN is event-driven and involve low-power consumption
- Uses a biological neuron model: Leaky-Integrate and Fire

$$\tau \frac{d}{dt}u(t) = -u(t) + RI(t)$$

$$I(t) = i_e(t) + \sum_f \sum_j \sum_k w_{jk}\, i_{jk}^{(f)}(t)$$

- A neuron fires a spike as soon as $u(t) = \theta$
- After spike, the membrane potential drops
- CNC: recurrently connected SNN where spike emission activity indicates DTN routing decisions



Vertices:

- Core neurons (yellow) represent *actions*
- Inhibitory neuron(s) (orange) provide potential regulation

Edges:

- Modulate the impact to the membrane potential of the arriving spikes

CNC: Learning and Exploitation

- Bootstrap by applying constant stimuli i_e to all neurons
- Synapse values (weights) influence the firing rates of the neurons
- The core neuron emitting spikes at the highest rate points to the selected action
- The "cost" C of the selected action is given by the expected bundle delivery time
- The CNC learns how to improve routing decisions by observing this cost (c_i: selected core neuron, g: inhibitory neuron): $\delta = \bar{C} - C$

$$\begin{aligned} w(c_j, c_i) &\leftarrow w(c_j, c_i) + \eta\delta \;\; ; j = 1, \ldots, n; i \neq j \\ w^k(g, c_i) &\leftarrow w^k(g, c_i) - \eta\delta \;\; ; k = 1, \ldots, n \end{aligned}$$

- Learning has linear time complexity
- Hyperparameter η is the learning factor

Reward Shaping

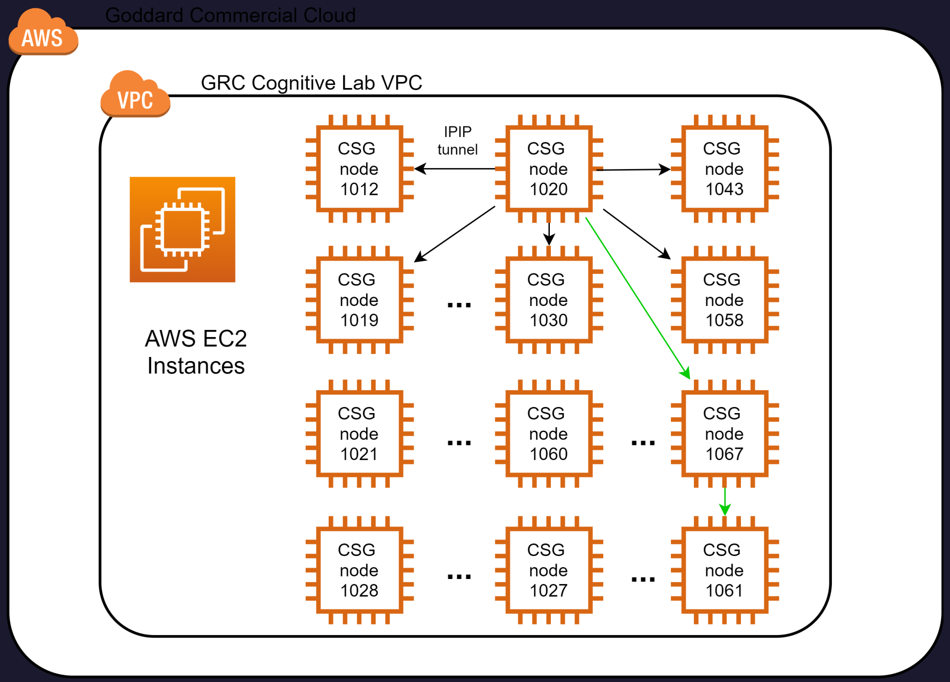
- Large propagation delays and link disruptions may prevent observing the cost *C* in practice
- Given a bundle addressed to node *d*, it is possible to estimate the cost at node *i* with:

$$c_{i,d} = T_{i,j} + T_{j,d} + D_{i,d}$$

$T_{i,j}$	Average delivery time to neighbor *j*: $T_{i,j} = (n_j + 1)S_j$ S_j is the service time (obtained cross-layer or via bundle ACK) n_j is the buffer occupancy
$T_{j,d}$	Average delivery time from neighbor *j* to destination *d*–passed by the neighbor
$D_{i,d}$	Stall time due to link disruptions obtained algorithmically from the contact plan

Experiments and Cloud Test Environment

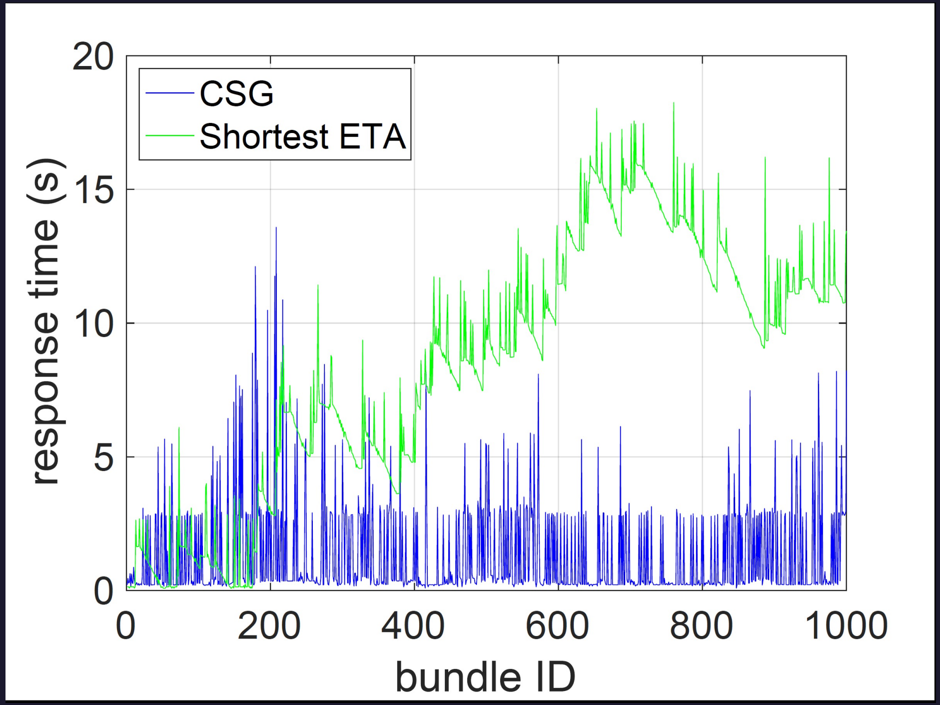
- Python implementation of the CSG (and CNC)
- Custom implementations:
  - Licklider Transmission Protocol (LTP)
  - Contact Graph Traversal: Shortest Estimate Time of Arrival (ETA)–similar to Contact Graph Routing
- Test data flow of 1000 bundles of 100 kB each
- Assumed no link disruptions as only 7% of links were affected
- Assumed links provide negligible packet loss rates except for link 1067–1061, which is affected by large signal loss yielding a packet loss ratio of 0.02.

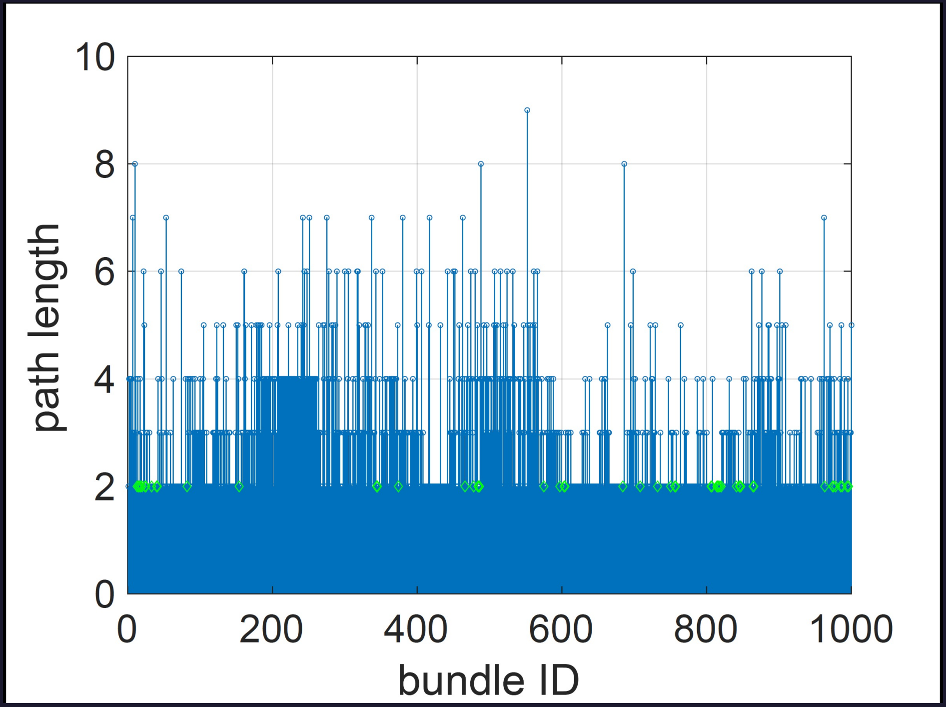


- Network consists of 12 Elastic Compute Cloud (EC2) instances
- T2.micro instance (1 CPU, 1 GB RAM)
- Ubuntu 18.04 image
- Limited to two network interfaces, so IPIP tunnels were used

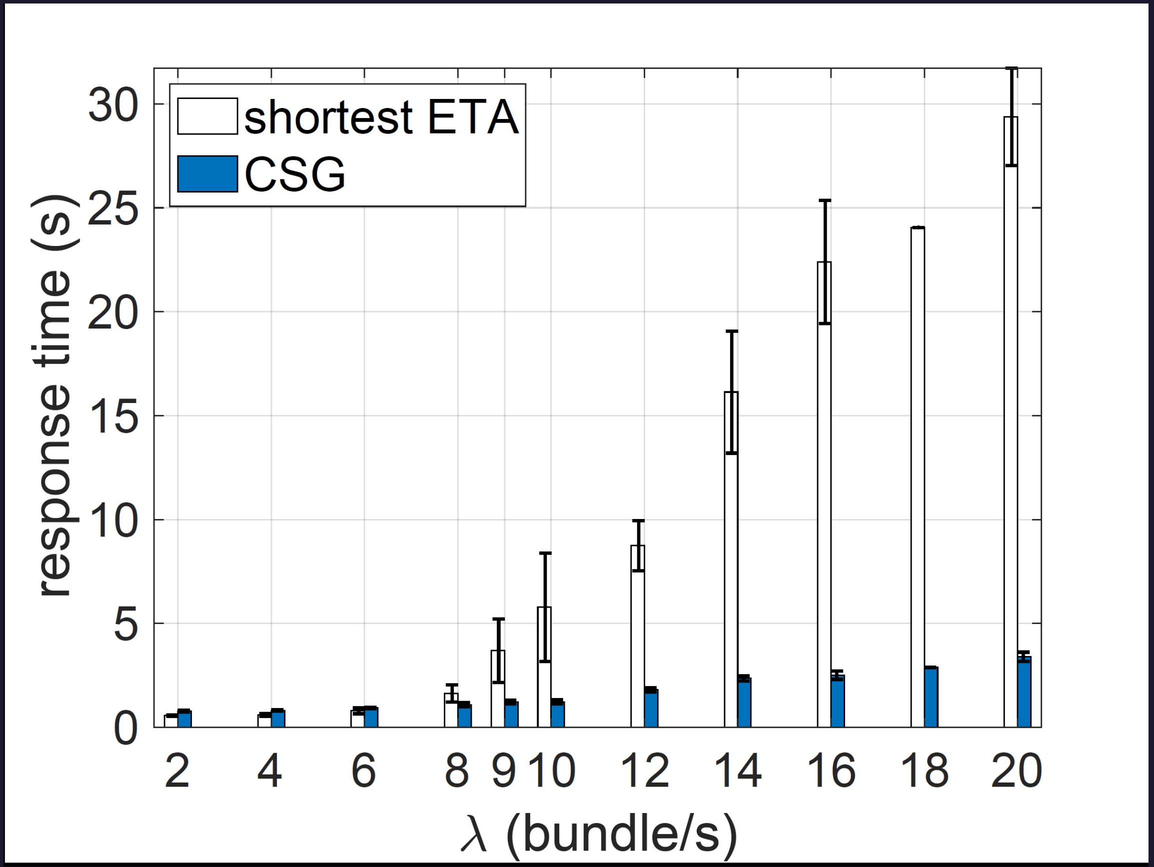
Time Series Observations

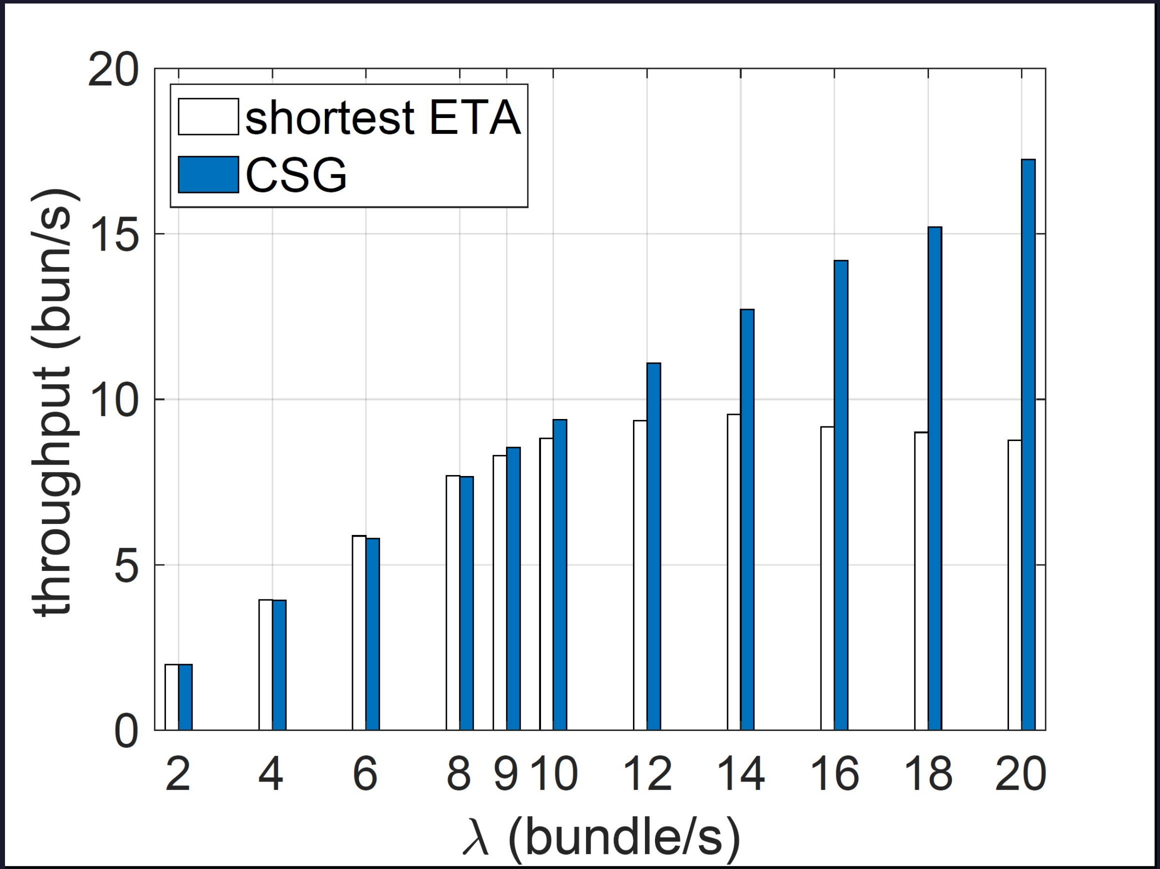
- Typical observations for a flow sent at 10 bundle/s
- Both the CSG and the Shortest ETA method attempt to minimize the bundle delivery time
- CSG dynamically learns to avoid link 1067–1061 (affected by higher loss)–the times when that link was used are highlighted (in green) in the path length chart



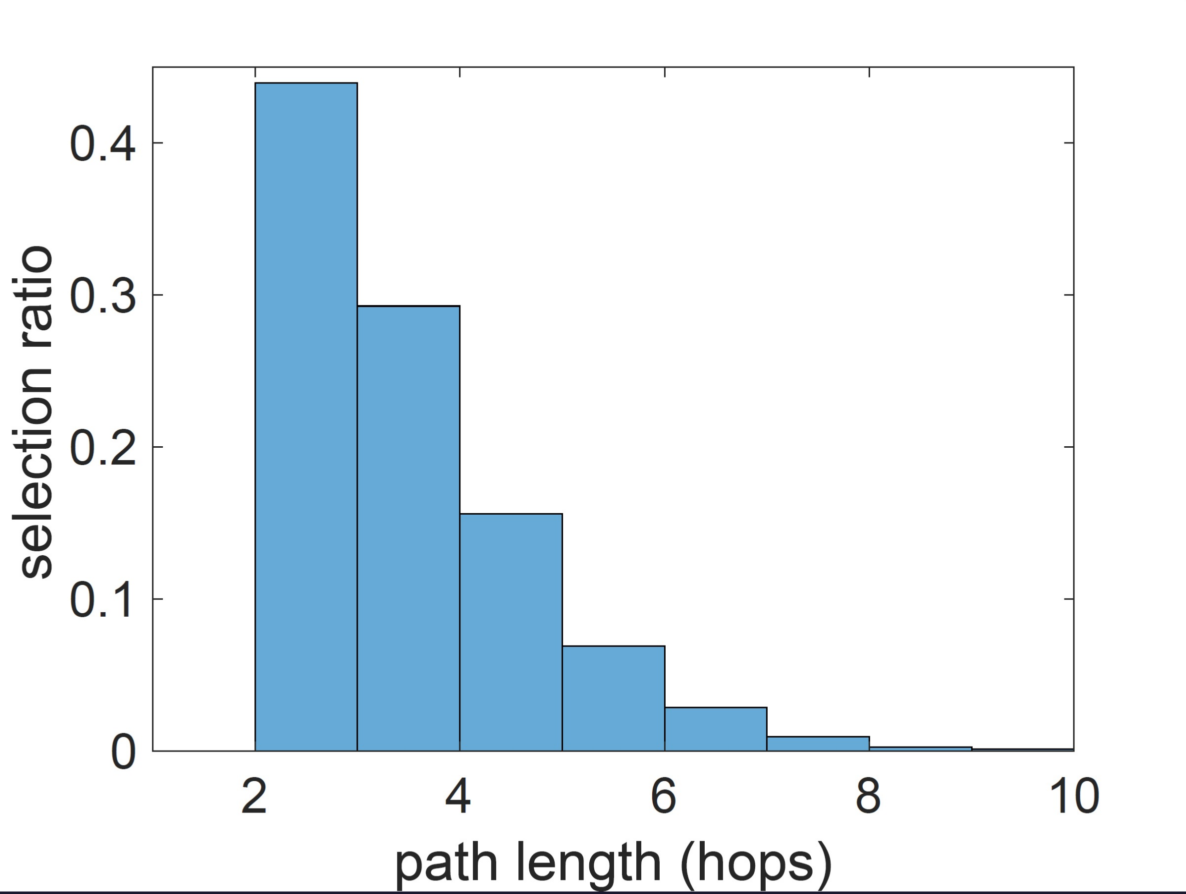


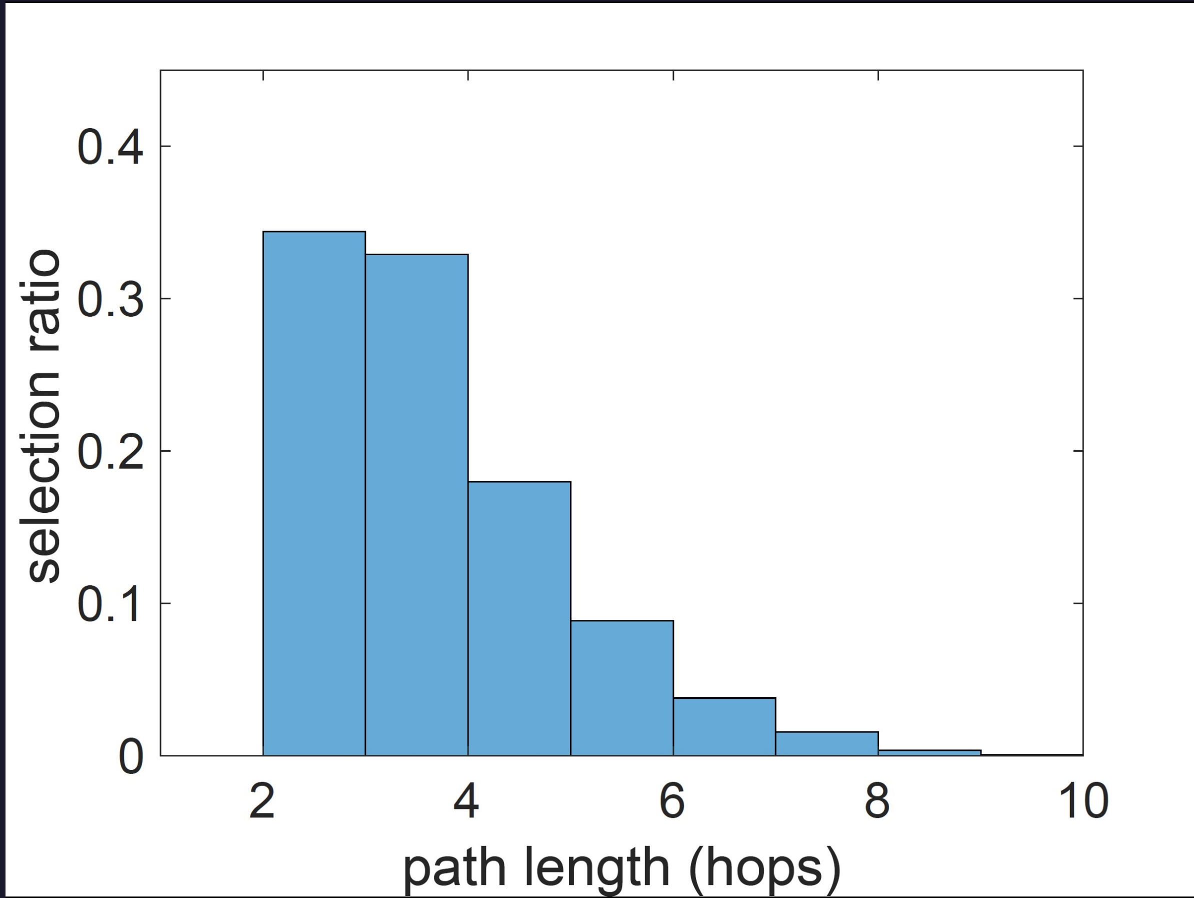
Average Trends





Selection of Path Lengths





Future Work

- Develop flight-like CSG implementation integrated within existing DTN framework
- Integrate software into multi-hop software defined radio ground testbed
- Experiment with increasingly dynamic topologies
- Enable discovery of neighboring nodes to support opportunistic routing

Thank You

References

- CubeSat Title Slide Image: https://www.nasa.gov/image-feature/three-cubesats-are-ejected-outside-the-kibo-laboratory-module-0
- SCaN Testbed Slide 1 Image: 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
- NASA Digital Transformation: https://www.nasa.gov/sites/default/files/atoms/files/396062_jan-jun_2019_it_talk_design_final.pdf