An AI-Enabled Testbed for Satellite Mega-Constellations

Systems and Protocol Aspects for Circumstellar Environments Research Group
SPACERG

Carleton University

Halim Yanikomeroglu
Gunes Karabulut Kurt
Pablo Madoery
Wafa Hasanain
Dhiraj Bhattacharjee
Jintao Liang
Ethan Fettes
Mohammed Mahyoub

NRC · CNRC

Colin Bellinger
Abhishek Naik
Sylvain Raymond

MDA

Stephane Martel
Khaled Ahmed
Guillaume Lamontagne

DRDC
RDDC

Sameera Siddiqui

Mitacs

Outline

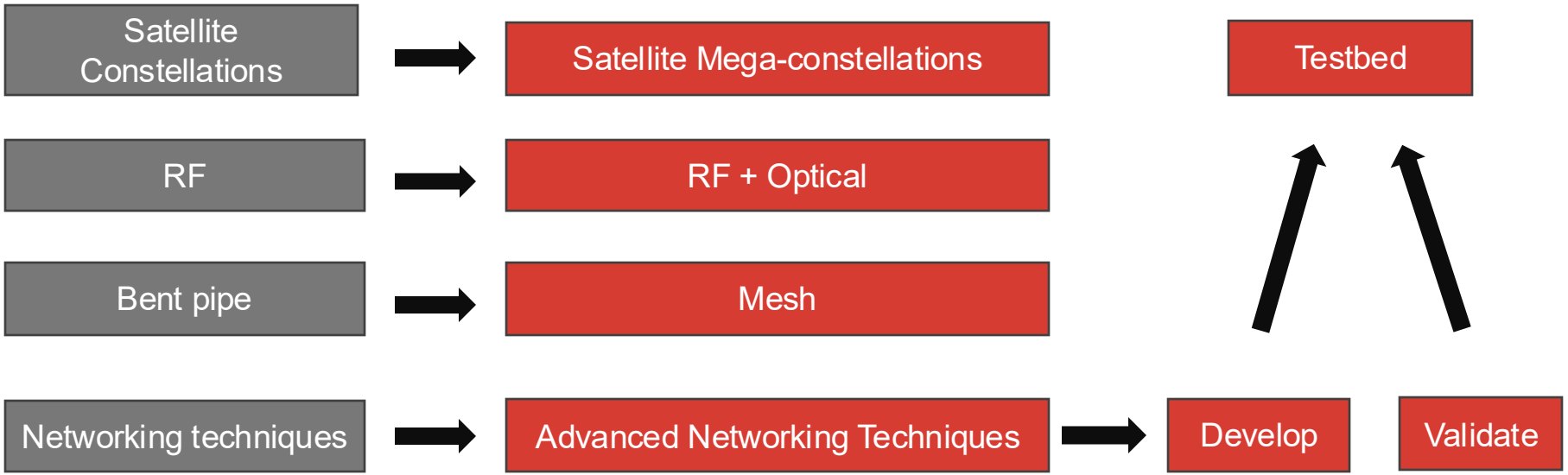
Why a testbed ?

What testbed ?

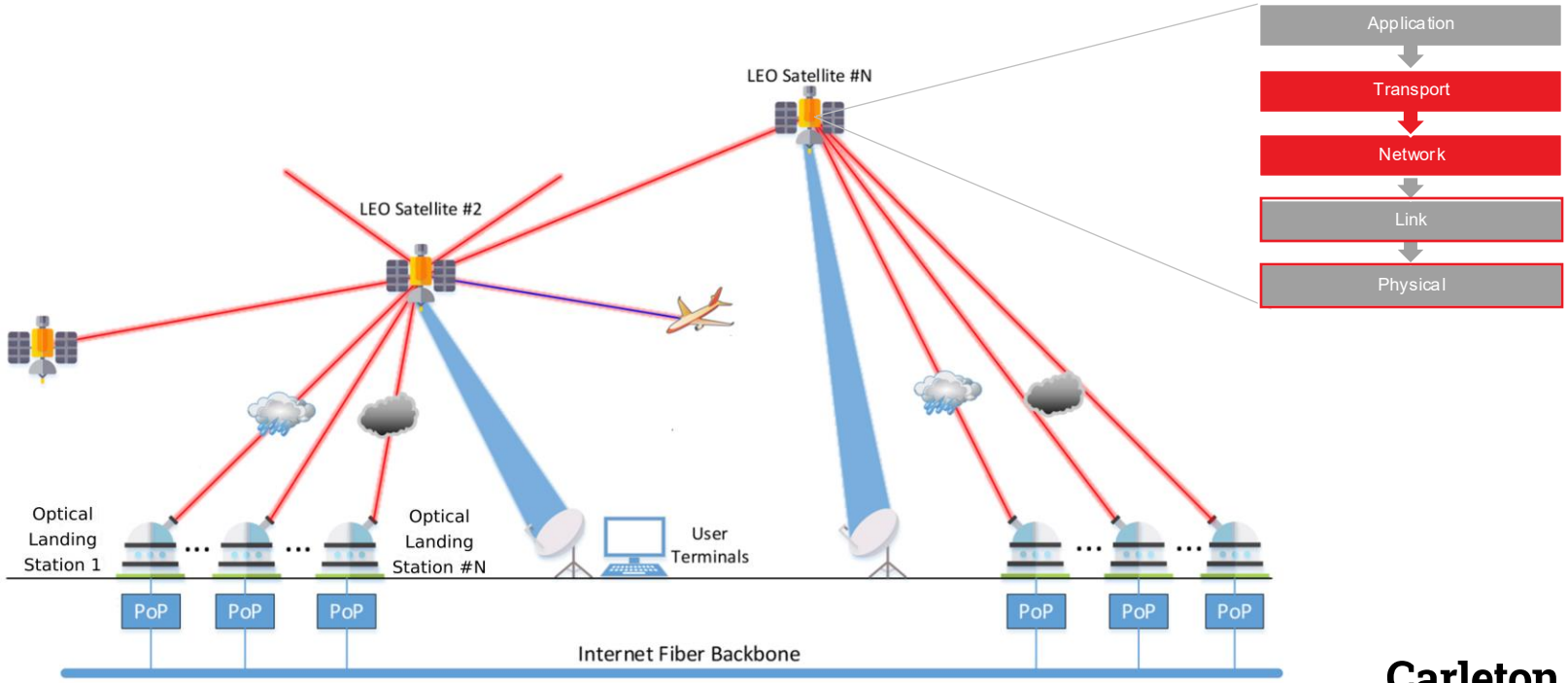
Demos & Research

Carleton University

Why a Testbed?



What Testbed ?



OSC system architecture proposed by MDA

Carleton University

What Testbed ?

Satellites

Landing Stations

User Terminals

Orbital Parameters

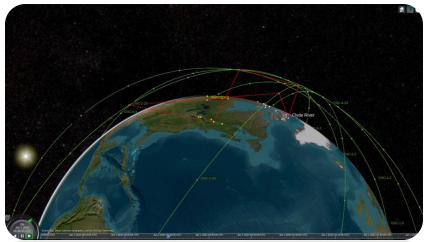
Geographical location

MATLAB

User Friendly GUI

Orbital Dynamics

Topology Computation



Keras PyTorch TensorFlow

Learning Dynamics

AI-Enabled Protocols & Architectures

OMNeT++ 6

Network Protocol Dynamics

Packet Level Simulation

Link Capacity

Node Capacity

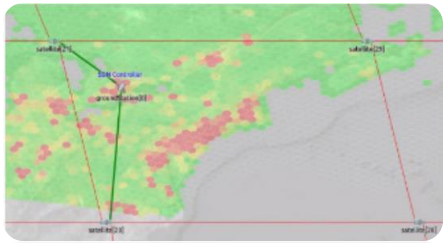
Traffic Demands

Weather Conditions

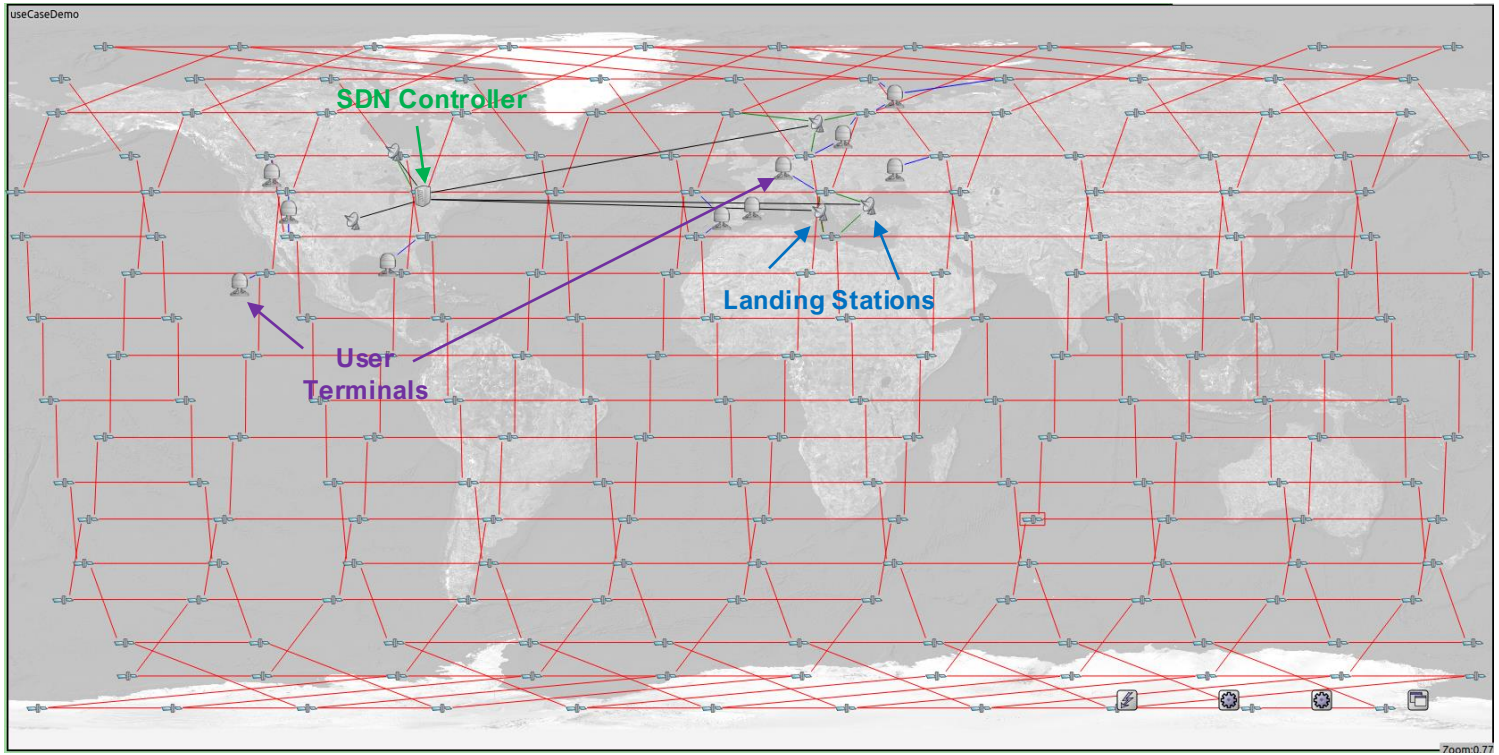
Routing Protocols

Transport Protocols

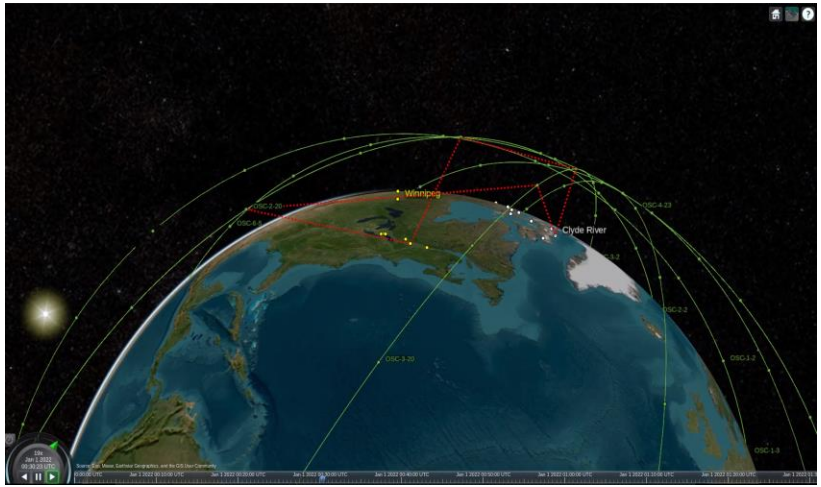
Automated Metrics Extraction

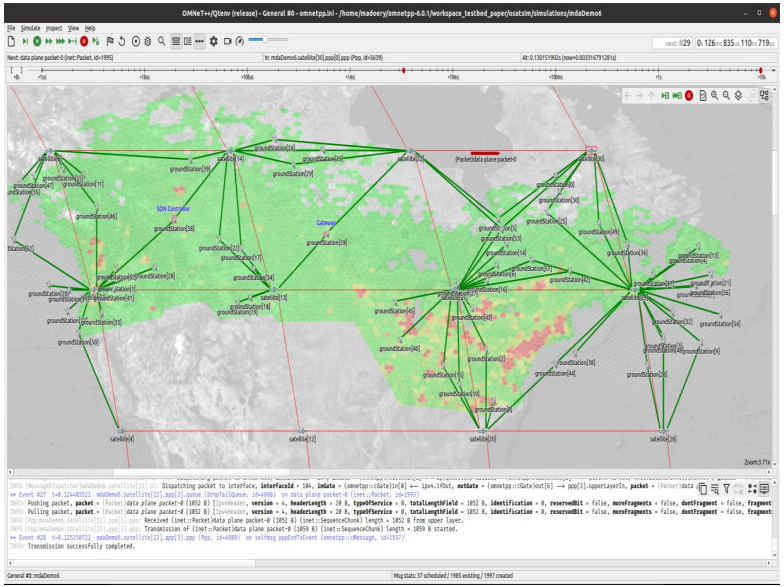


What Testbed ?



Demo 1:





INET Framework - Link

OS³ - The Open Source Satellite Simulator - Link

Atlas: OMNeT++ and INET framework for simulation Low-Earth Orbit Satellite Constellations - Link

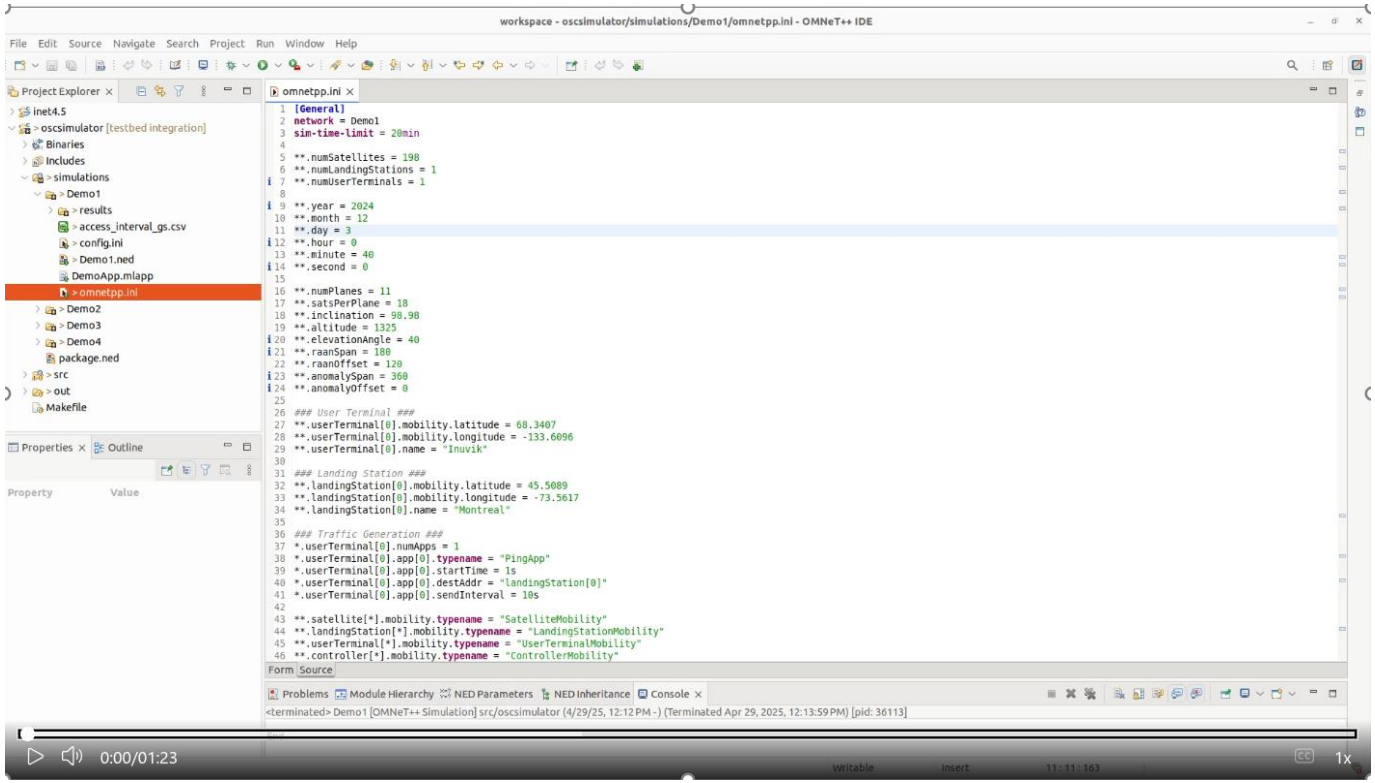
OpenFlow Extension for INET Framework - Link

H3 Library - Link

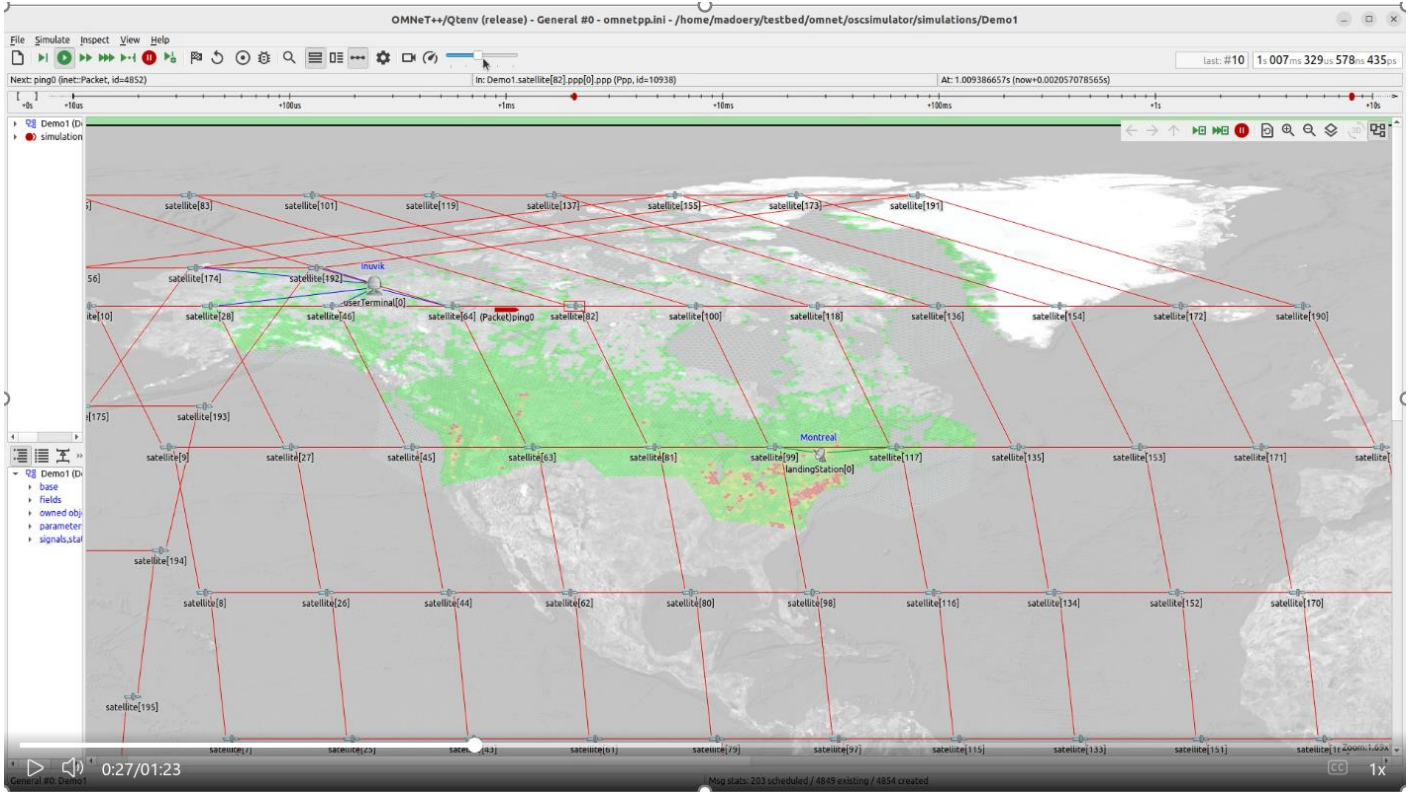
The Kontur Population dataset - Link

Meteoblue weather data - Link

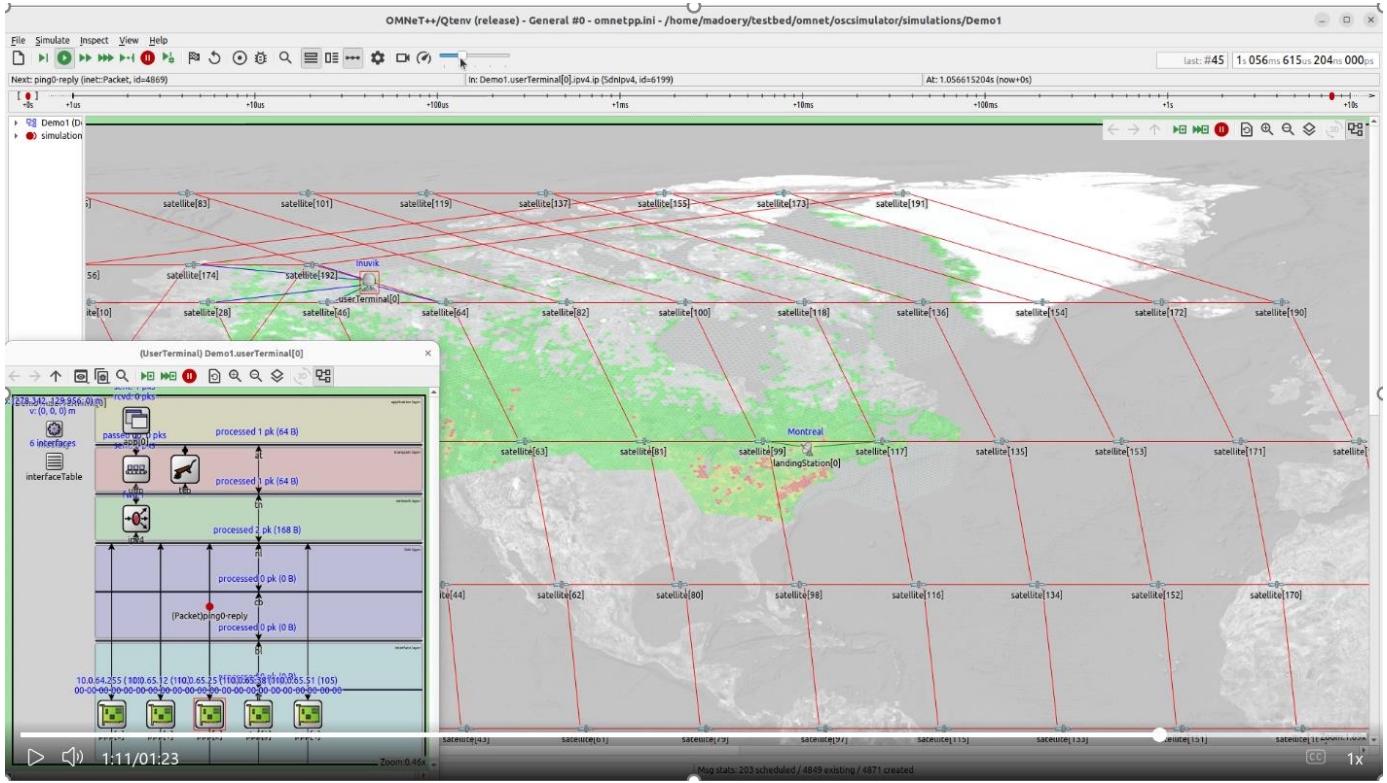
Demo 1:


workspace - oscsimulator/simulations/Demo1/omnetpp.ini - OMNeT++ IDE
File Edit Source Navigate Search Project Run Window Help
Project Explorer
inet4.5
oscsimulator [testbed integration]
Binaries
Includes
simulations
Demo1
results
access_interval_gs.csv
config.ini
Demo1.ned
DemoApp.mlapp
omnetpp.ini
Demo2
Demo3
Demo4
package.ned
src
out
Makefile
Properties
Outline
Property
Value
omnetpp.ini
[General]
network = Demo1
sim-time-limit = 20min
**.numSatellites = 198
**.numLandingStations = 1
**.numUserTerminals = 1
**.year = 2024
**.month = 12
**.day = 3
**.hour = 0
**.minute = 40
**.second = 0
**.numPlanes = 11
**.satsPerPlane = 18
**.inclination = 98.98
**.altitude = 1325
**.elevationAngle = 40
**.raanSpan = 180
**.raanOffset = 120
**.anomalySpan = 360
**.anomalyOffset = 0
User Terminal
**.userTerminal[0].mobility.latitude = 68.3407
**.userTerminal[0].mobility.longitude = -133.6096
**.userTerminal[0].name = "Inuvik"
Landing Station
**.landingStation[0].mobility.latitude = 45.5089
**.landingStation[0].mobility.longitude = -73.5617
**.landingStation[0].name = "Montreal"
Traffic Generation
*.userTerminal[0].numApps = 1
*.userTerminal[0].app[0].typename = "PingApp"
*.userTerminal[0].app[0].startTime = 1s
*.userTerminal[0].app[0].destAddr = "landingStation[0]"
*.userTerminal[0].app[0].sendInterval = 10s
**.satellite[*].mobility.typename = "SatelliteMobility"
**.landingStation[*].mobility.typename = "LandingStationMobility"
**.userTerminal[*].mobility.typename = "UserTerminalMobility"
**.controller[*].mobility.typename = "ControllerMobility"
Form Source
Problems Module Hierarchy NED Parameters NED Inheritance Console
<terminated> Demo1 [OMNeT++ Simulation] src/oscsimulator (4/29/25, 12:12 PM -) (Terminated Apr 29, 2025, 12:13:59 PM) [pid: 36113]
0:00/01:23
1x

Demo 1:

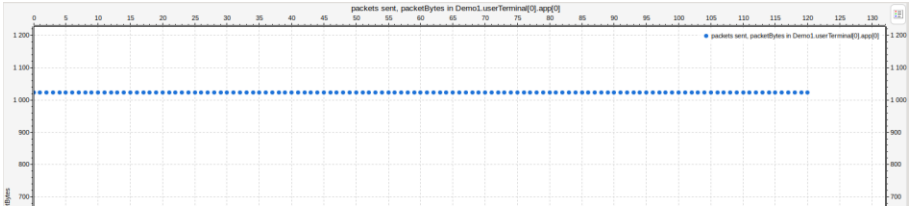

OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - /home/madoery/testbed/omnet/oscsimulator/simulations/Demo1
File Simulate Inspect View Help
Inuvik
userTerminal[0]
Montreal
landingStation[0]
0:27/01:23
1x

Demo 1:

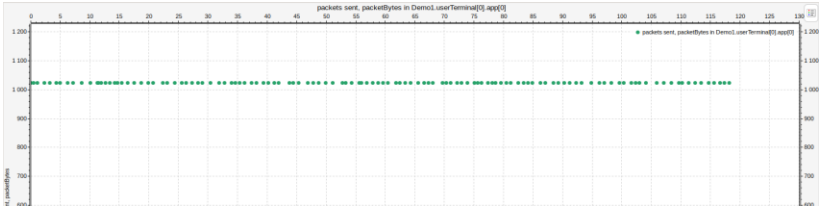

OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - /home/madoery/testbed/omnet/oscsimulator/simulations/Demo1
File Simulate Inspect View Help
Inuvik
Montreal
(UserTerminal) Demo1.userTerminal[0]
processed 1 pk (64 B)
processed 1 pk (64 B)
processed 2 pk (168 B)
processed 0 pk (0 B)
processed 0 pk (0 B)
1:11/01:23
1x

Demo 1:

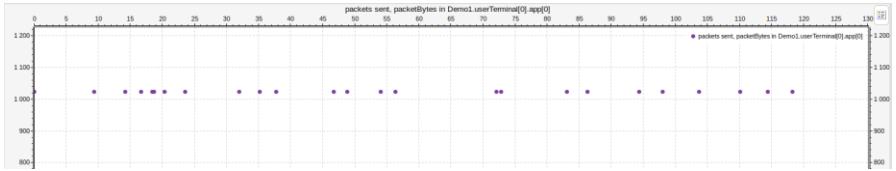
Constant Bit Rate (CBR)



Normal Distribution

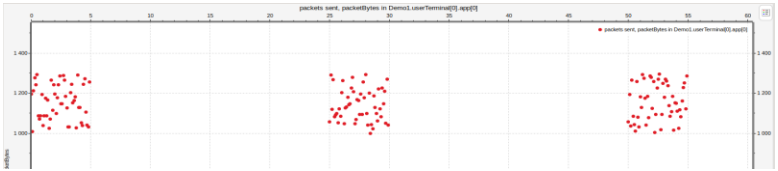


Exponential Distribution




```
*.userTerminal[0].numApps = 1
*.userTerminal[0].app[0].typename = "UdpBasicApp"
*.userTerminal[0].app[0].destAddresses = "landingStation[0]"
*.userTerminal[0].app[0].destPort = 5000
*.userTerminal[0].app[0].messageLength = 1024B
*.userTerminal[0].app[0].sendInterval = exponential(5s)
```

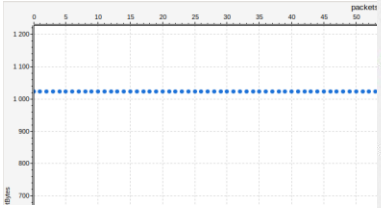
Traffic Burst



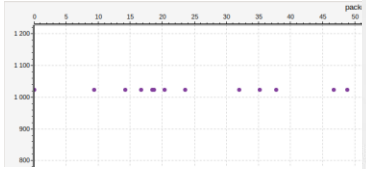

```
*.userTerminal[0].app[0].destPort = 5000
*.userTerminal[0].app[0].messageLength = intuniform(1000B, 1300B)
*.userTerminal[0].app[0].burstDuration = 5s
*.userTerminal[0].app[0].sleepDuration = 20s
*.userTerminal[0].app[0].sendInterval = 0.1s
```

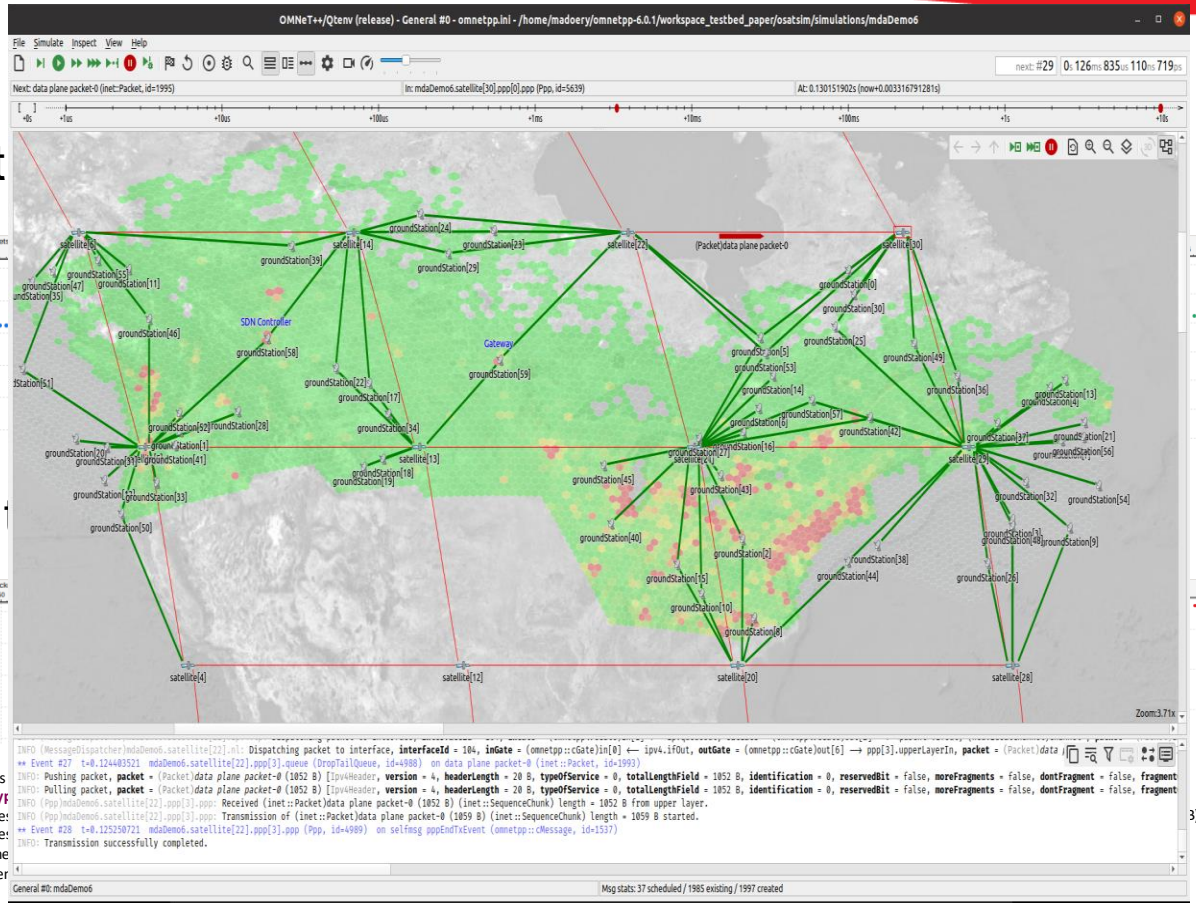
Demo 1:

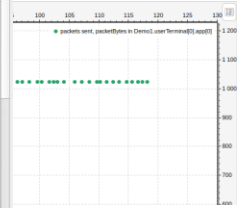
Constant

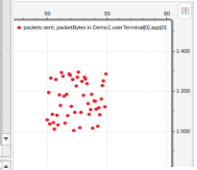


Exponent





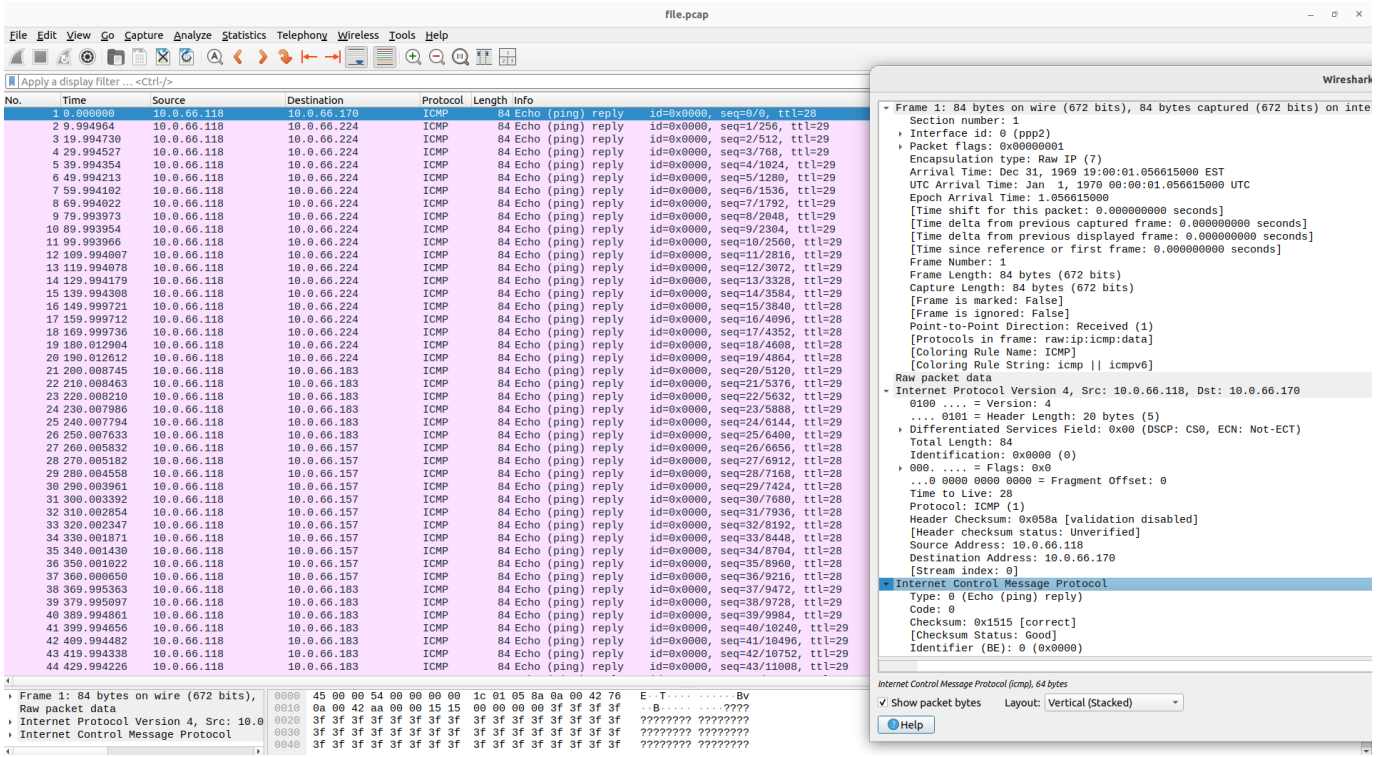




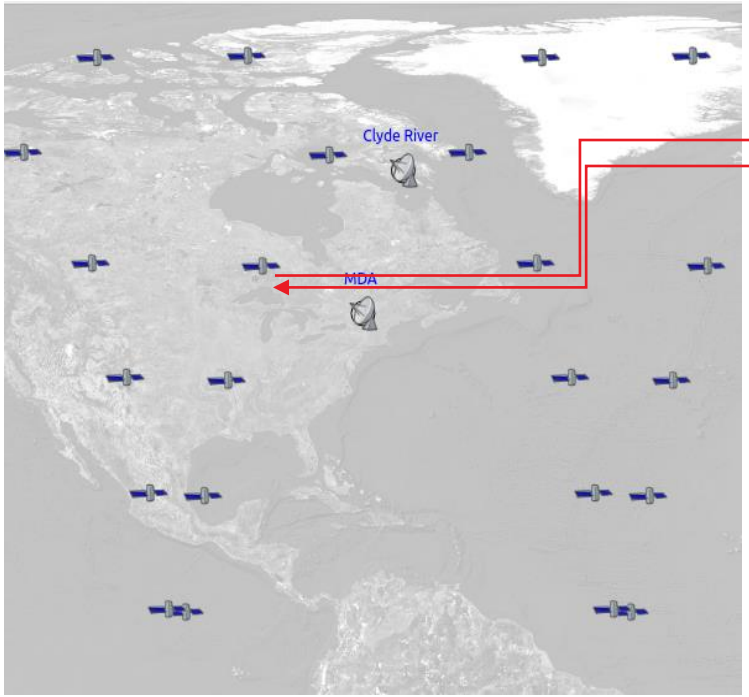
*.userTerminal[0].numApps
*.userTerminal[0].app[0].typ
*.userTerminal[0].app[0].des
*.userTerminal[0].app[0].des
*.userTerminal[0].app[0].me
*.userTerminal[0].app[0].ser

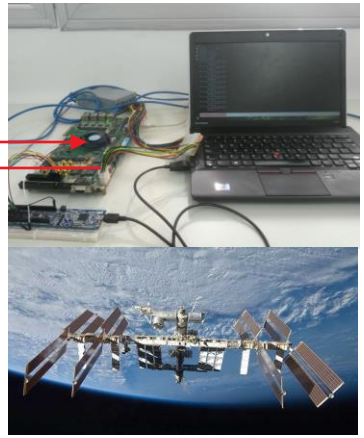
Demo 1:

Wireshark



Demo 1:





Omnet++ supports emulation by providing modules that act as bridges between the simulated and real domains

Possible setups:

simulated node in a real network

a simulated subnet in real network

real-world node in simulated network

simulated protocol in a real network node

real application in a simulated network node

A roadmap to Digital Twins!

Demo 2: QoS-aware Routing & Queue Management

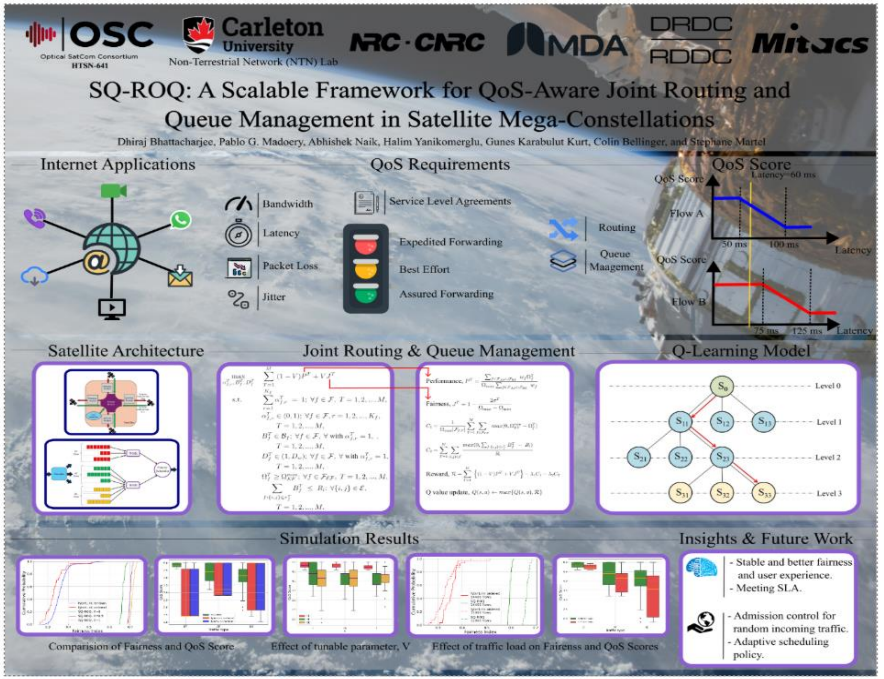
Dhiraj Bhattacharjee

SQ-ROQ: A Scalable Framework for QoS-Aware Joint Routing and Queue Management in Satellite Mega-Constellations

Dhiraj Bhattacharjee[1], Pablo G. Madoery[1], Abhishek Naik[3], Halim Yanikomerglu[1], Güneş Karabulut Kurt[2], Colin Bellinger[3], and Stéphane Martel[4]

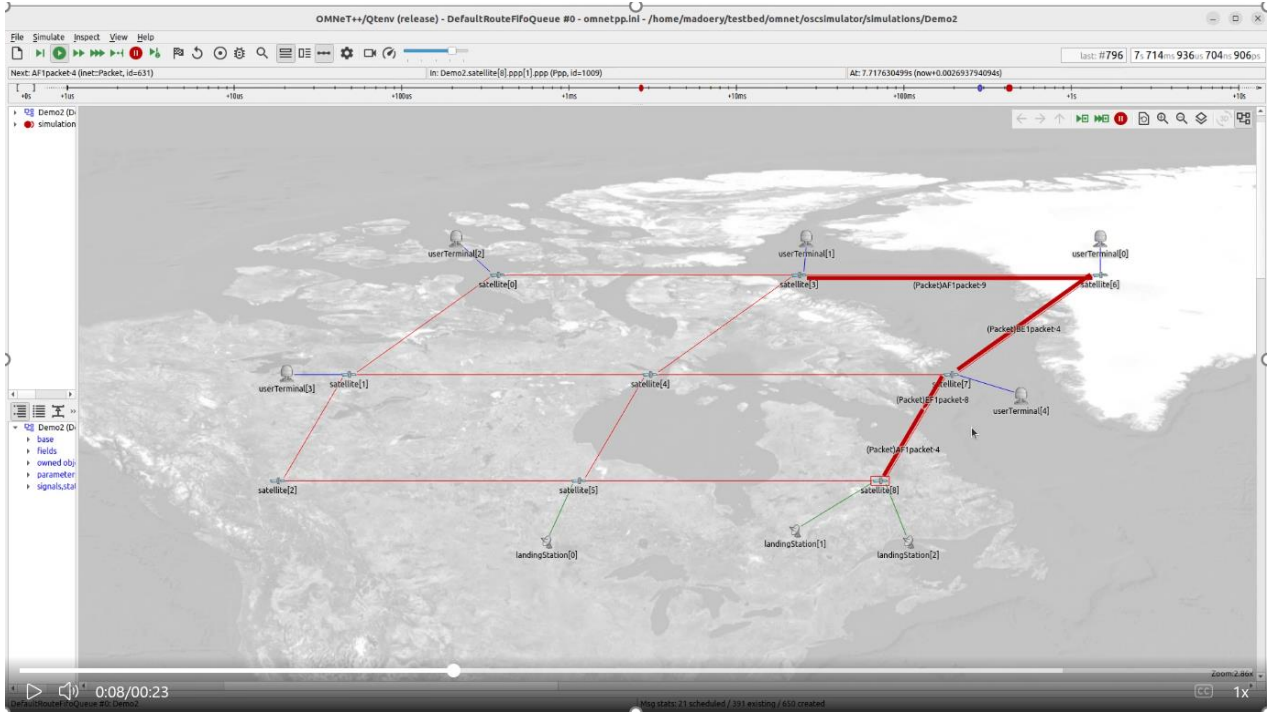
[1]Department of Systems and Computer Engineering, Carleton University, Canada
[2]Polytechnique Montreal, Canada
[3]National Research Council, Canada
[4]Satellite Systems, MDA, Canada

***Abstract*—The modern Internet accommodates a wide range of applications with heterogeneous quality of service (QoS) requirements across multiple network performance metrics. Low Earth orbit (LEO) satellite constellations have emerged as a promising solution to support these diverse services, not only in rural and remote areas but also in urban environments as a complement to terrestrial networks. Ensuring QoS compliance in such networks necessitates the joint optimization of routing and queue management, as effective traffic handling is critical to maintaining performance guarantees across multiple flows. In such as surveillance, defense, computing, sensing, and navigation with ubiquitous coverage, reliability, and sustainability. Achieving these ambitious goals affordably requires a pivotal consideration: the seamless integration of terrestrial and non-terrestrial networks (NTN) [1]. NTNs are gaining significant attention from space industry leaders such as SpaceX, Telesat, Amazon, and OneWeb, and are also drawing increasing interest from the research community. Low Earth orbit (LEO) satellite's proximity to Earth offers significant advantages over**



Submitted to IEEE Transactions on Network and Service Management

Demo 2: QoS-aware Routing & Queue Management



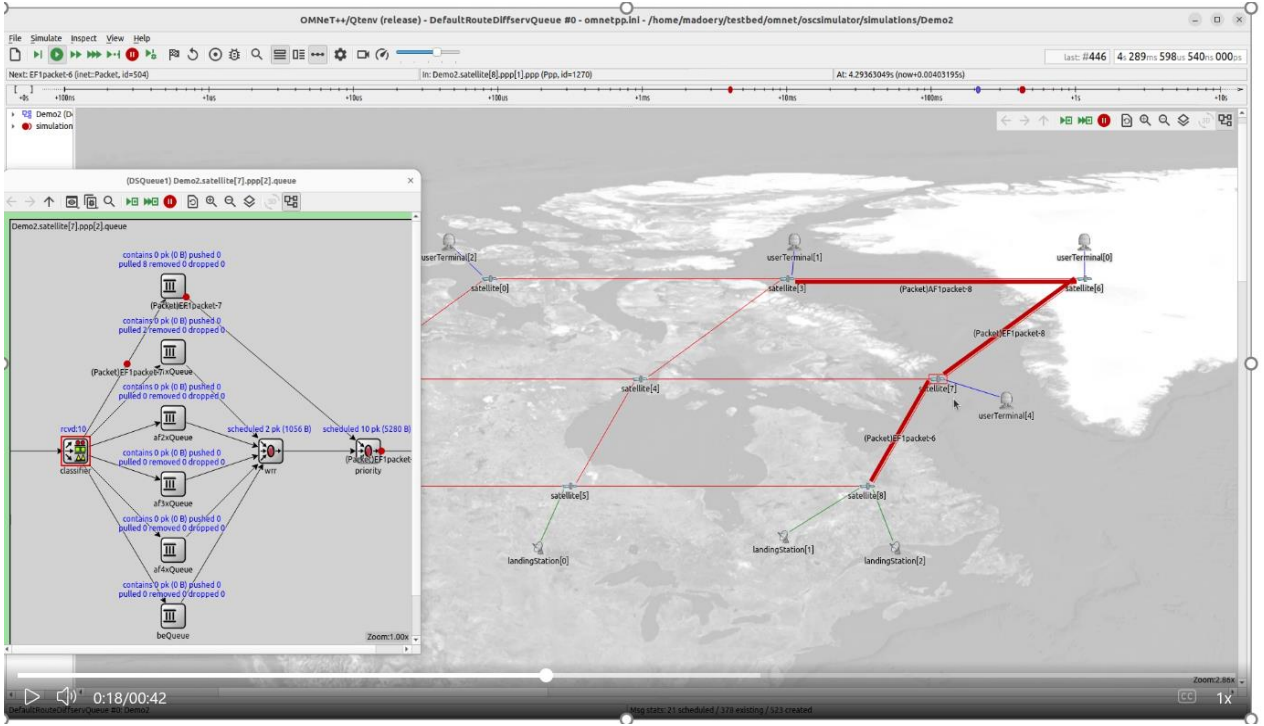
RFC 2474 Defines the *Differentiated Services (DiffServ)* architecture

Expedited Forwarding (EF)

Assured Forwarding (AF)

Best Effort (BE)

Demo 2: QoS-aware Routing & Queue Management



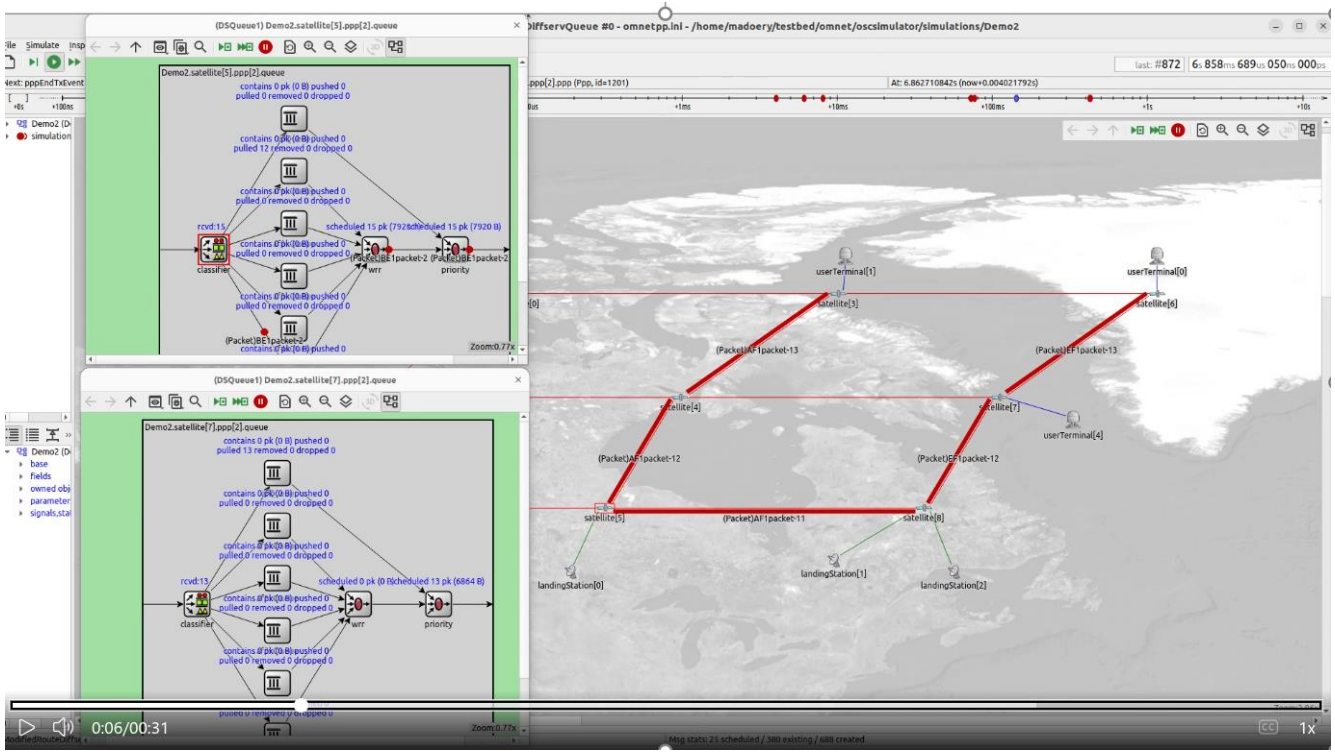
RFC 2474 Defines the *Differentiated Services (DiffServ)* architecture

Expedited Forwarding (EF)

Assured Forwarding (AF)

Best Effort (BE)

Demo 2: QoS-aware Routing & Queue Management



RFC 2474 Defines the *Differentiated Services (DiffServ)* architecture

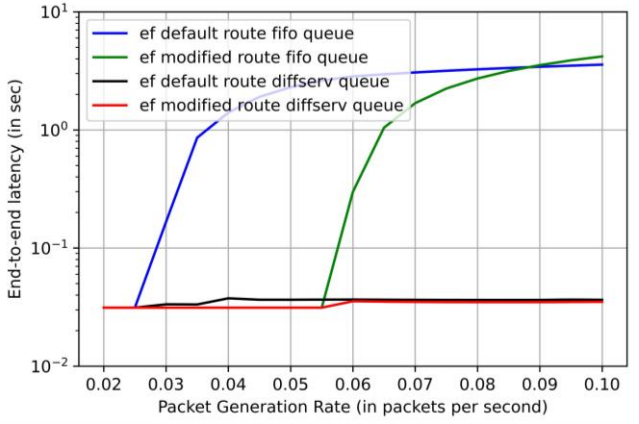
Expedited Forwarding (EF)

Assured Forwarding (AF)

Best Effort (BE)

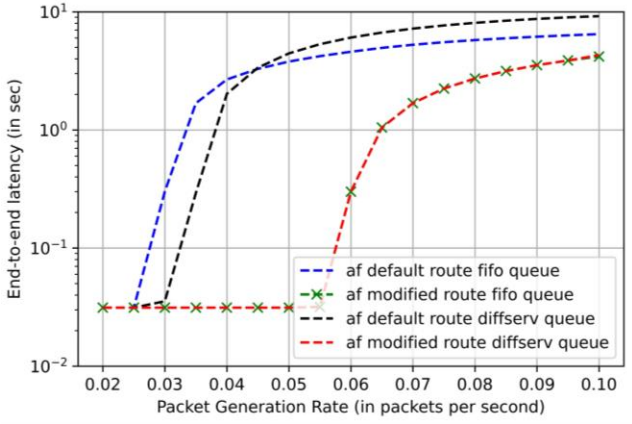
Demo 2: QoS-aware Routing & Queue Management

Expedited Forwarding (EF)



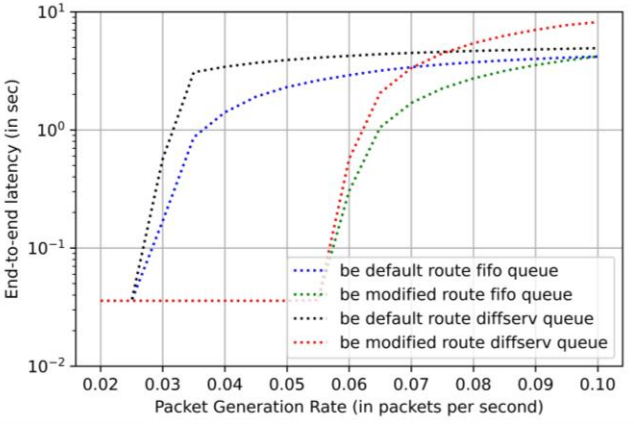
Changing route gives some advantage upto a point

Assured Forwarding (AF)



We need the queueing architecture to see the real gain here

Best Effort (BE)



We are paying the cost here

Fig. 1: Average end-to-end latency vs Traffic Load for EF, AF, and BE traffic

Research Question: How to select **routes** and **weights** to improve **QoS** and **fairness**?

Demo 3: Software-Defined Network Architectures

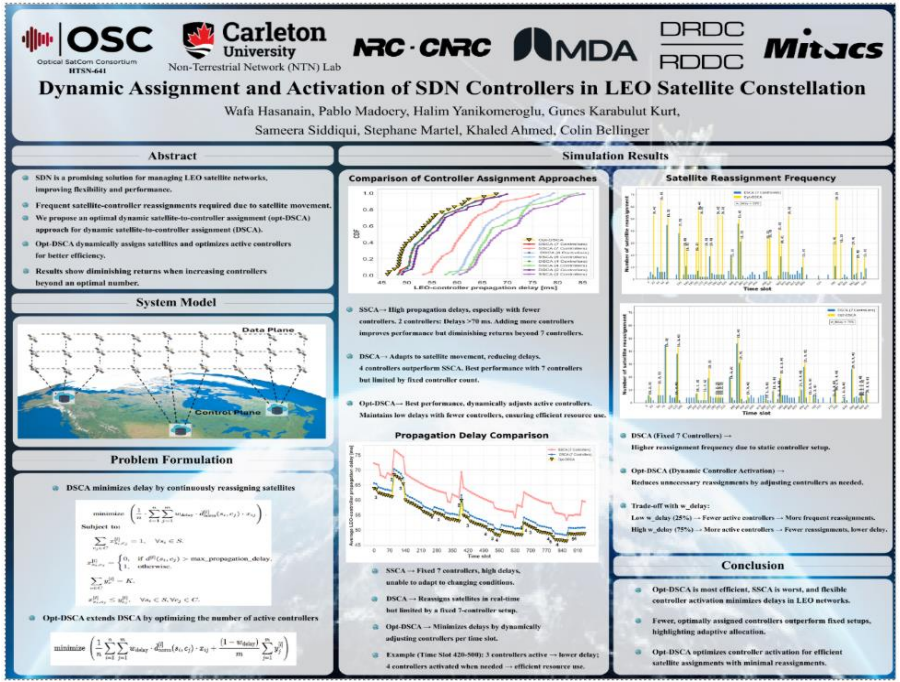
Wafa
Hasanain

Dynamic Assignment and Activation of SDN Controllers in LEO Satellite Constellations

Wafa Hasanain*, Pablo G. Madoery*, Halim Yanikomeroglu*, Gunes Karabulut Kurt*‖, Sameera Siddiqui§, Stephane Martel‡, Khaled Ahmed‡, Colin Bellinger†

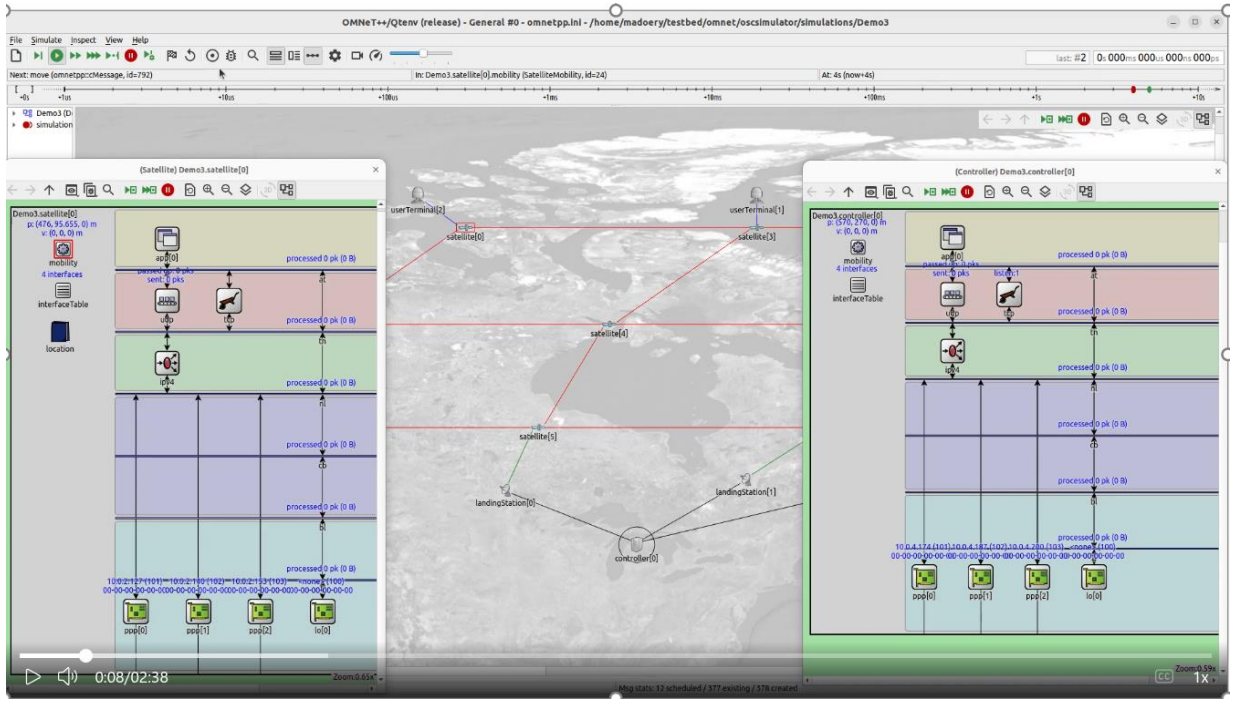
* *Non-Terrestrial Networks (NTN) Lab, Department of Systems and Computer Engineering, Carleton University, Canada*
‖ *Poly-Grames Research Center, Department of Electrical Engineering, Polytechnique Montréal, Montréal, Canada*
§ *Defence Research and Development Canada, Canada*
‡ *Satellite Systems, MDA, Canada*
† *National Research Council Canada, Canada*

***Abstract*—Software-defined networking (SDN) has emerged as a promising approach for managing traditional satellite communication. This will enhance opportunities for future services, including integrating satellite and terrestrial networks. In this paper, we develop an SDN framework using the OMNeT++ simulator to model the behavior of SDN controllers and switches in non-terrestrial networks (NTN). Dynamic controller assignment is one of the most significant challenges for large low Earth orbit (LEO) constellations. Due to the movement of LEO satellites, satellite-controller assignments must be updated frequently to maintain low propagation delays. To address this issue, we present a dynamic satellite-to-controller assignment (DSCA) optimization problem that continuously adjusts these** its benefits, SDN also presents new challenges when applied to NTN. The controller assignment problem is a significant challenge that is critical in large-scale distributed networks. This problem involves determining the optimal number of controllers required for efficient network management and assigning satellites (or switches) to controllers to maximize network resilience and quality of service (QoS) [2]. Satellite networks with dynamic topologies add another level of complexity. Since satellites orbit the Earth rapidly, their relative positions to SDN controllers change constantly, resulting in fluctuating communication latencies. As a result, satellite-to-



Accepted at IEEE PIMRC Conference and under review at IEEE Open Journal of Vehicular Technology

Demo 3: Software-Defined Network Architectures



Research Questions:

How many controllers and where?

How to activate them dynamically over time?

How to associate satellites to controllers over time?

What are the advantages and what are the penalties?

How to balance performance and overhead?

Demo 4: Weather-aware Routing

Ethan Fettes

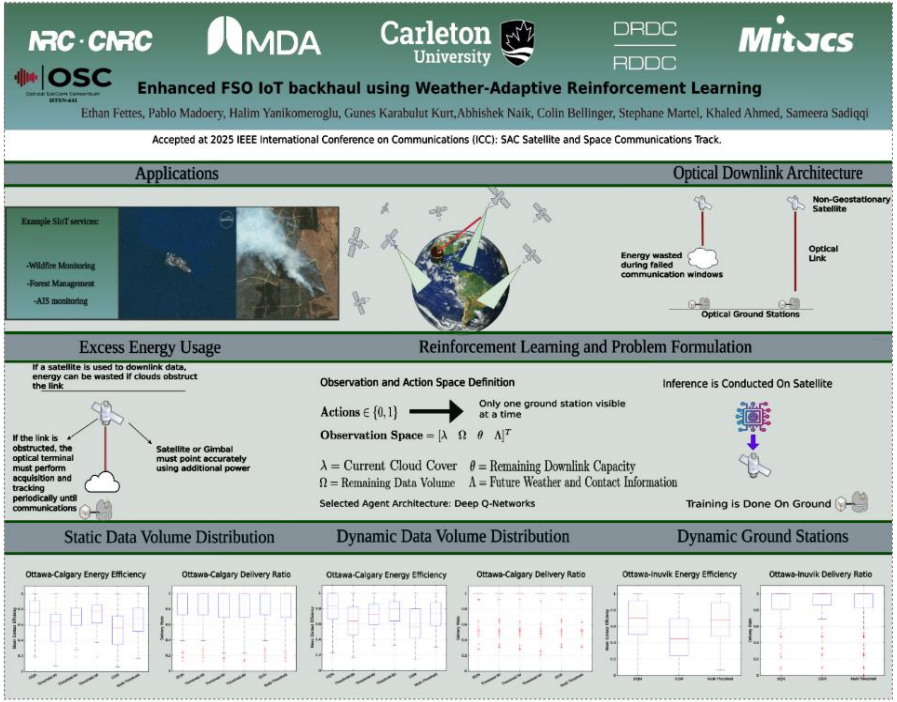
Energy-Efficient Satellite IoT Optical Downlinks Using Weather-Adaptive Reinforcement Learning

Ethan Fettes*, Pablo G. Madoery*, Halim Yanikomeroglu*, Gunes Karabulut Kurt*‖, Abhishek Naik †, Colin Bellinger†, Stéphane Martel§, Khaled Ahmed§, Sameera Siddiqui‡

* Non-Terrestrial Networks (NTN) Lab, Department of Systems and Computer Engineering, Carleton University, Canada
‖ Poly-Grames Research Center, Department of Electrical Engineering, Polytechnique Montréal, Montréal, Canada
† National Research Council Canada, Canada
‡ Defence Research and Development Canada, Canada
§ Satellite Systems, MDA, Canada

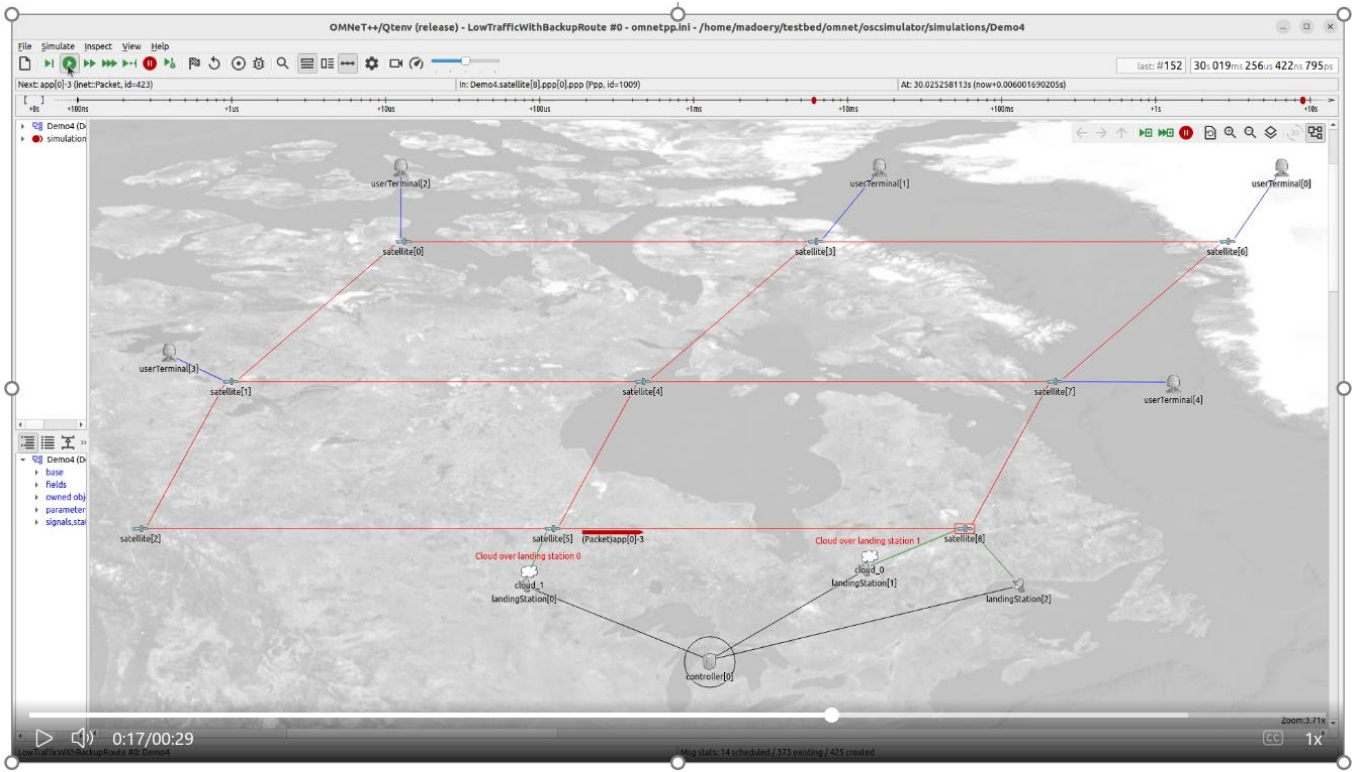
***Abstract*—Internet of Things (IoT) devices have become increasingly ubiquitous with applications not only in urban areas but remote areas as well. These devices support industries such as agriculture, forestry, and resource extraction. Due to the device location being in remote areas, satellites are frequently used to collect and deliver IoT device data to customers. As these devices become increasingly advanced and numerous, the amount of data produced has rapidly increased potentially straining the ability for radio frequency (RF) downlink capacity. Free space optical**

In recent years free space optical (FSO) communications have been proposed due to the high data rates that can be achieved. However, FSO communications are heavily impacted by adverse weather conditions such as cloud cover. As cloud cover can be forecasted using physical weather models such as the regional deterministic prediction system (RDPS) [2] or machine learning-based techniques [3], using this information to make intelligent routing decisions is possible.



Accepted at IEEE PIMRC and the IEEE International Conference on Communications (ICC 2025)

Demo 4: Weather-aware Routing



Research Questions:

How many and which backup routes to configure?

What are the advantages and what are the penalties?

How to balance performance and overhead?

How to use weather forecasts and Reinforcement Learning to configure backup routes?

Green Satellite Networks

Jintao Liang

Green Traffic Engineering for Satellite Networks Using Segment Routing Flexible Algorithm

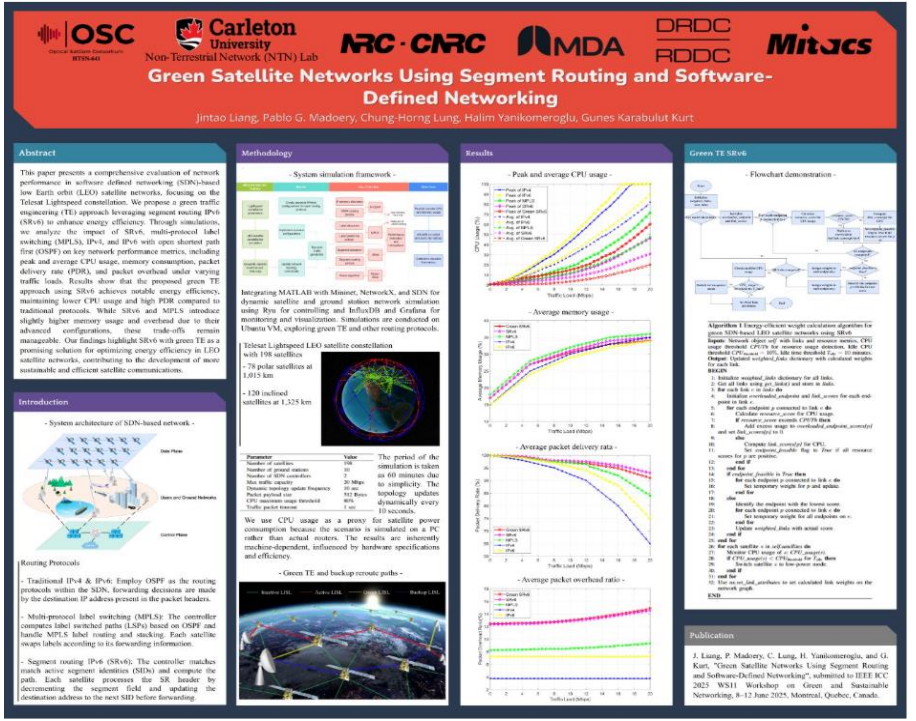
Jintao Liang*, Pablo G. Madoery*, Chung-Horng Lung†, Halim Yanikomeroglu*, Gunes Karabulut Kurt*‡

* Non-Terrestrial Networks (NTN) Lab, Department of Systems and Computer Engineering, Carleton University, Canada
†Department of Systems and Computer Engineering, Carleton University, Canada
‡ Poly-Grames Research Center, Department of Electrical Engineering, Polytechnique Montréal, Montréal, Canada

***Abstract*—Large-scale low-Earth-orbit (LEO) constellations demand routing that simultaneously minimizes energy, guarantees delivery under congestion, and meets latency requirements for time-critical flows. We present a segment routing over IPv6 (SRv6) flexible algorithm (Flex-Algo) framework that consists of three logical slices: an energy-efficient slice (Algo 130), a high-reliability slice (Algo 129), and a latency-sensitive slice (Algo 128). The framework provides a unified mixed-integer linear program (MILP) that combines satellite CPU power, packet delivery rate (PDR), and end-to-end latency into a single objective, allowing a lightweight software-defined network (SDN) controller to steer traffic from the source node. Emulation of Telesat's Lightspeed constellation shows that, compared with different routing schemes, the proposed design reduces the average CPU usage by 73%, maintains a PDR above 91% during traffic bursts, and decreases urgent flow delay by 18 ms between Ottawa and Vancouver. The results confirm Flex-Algo's value as a slice-based enables dynamic control over the network, supporting green TE for energy-constrained satellite constellations.**

Flexible algorithm (Flex-Algo) represents a pivotal innovation in SR, extending interior gateway protocols (IGP) to compute constraint-based paths within the control plane [8]. Flex-Algo can create multiple virtual topologies based on different real-time metrics, such as latency and energy consumption, and dynamically select paths at the source node once the SID stack is assigned. The inherent flexibility of Flex-Algo simplifies network management, enhances scalability, and supports intent-based networking frameworks [9], making it particularly suitable for satellite networks that must rapidly adapt to link failures, load fluctuations, and energy constraints.

This paper proposes a comprehensive framework that inte-



Accepted at IEEE Global Communications Conference (Globecom 2025)

Thank you :)

Carleton University