Electrical Flow Routing for LEO Satellite Mega-Constellations: A Circuit-Inspired Approach

Nicholas Zayfman

nzayfma1@jh.edu

I. Introduction

The landscape of satellite communications is undergoing a comprehensive transformation. Within the past five years, Low Earth Orbit (LEO) mega-constellations have evolved from ambitious concepts to operational reality. As of November 2025, SpaceX's Starlink constellation consists of 7,788 satellites in LEO, comprising approximately 53% of all active satellites, while competitor OneWeb has deployed 648 operational satellites. [1] These constellations promise global broadband coverage, reduced latency compared to traditional geostationary satellites, and unprecedented network capacity. However, this explosive growth introduces significant technical challenges, particularly in the domain of network routing and traffic management.

Unlike traditional satellite systems that rely primarily on ground-based routing through gateway stations, modern LEO constellations employ inter-satellite links (ISLs) using Free-Space Optical (FSO) communication. [2] This creates a dynamic, three-dimensional mesh network where satellites constantly move relative to each other at orbital velocities exceeding 7 km/s dependent on altitude. [3] The resulting network topology changes continuously as satellites rise and set relative to ground stations, orbital planes shift, and link geometries evolve. Recent research has highlighted significant challenges in routing for these dynamic networks, including frequent route updates due to high-speed satellite motion, load imbalance from uneven traffic distribution, and the need for adaptive algorithms that can respond to changing network conditions. Managing traffic routing in such an environment requires algorithms that can handle not only the scale of thousands of network nodes, but also the inherent dynamics of the system.

Traditional routing protocols, designed for terrestrial networks with static topologies, struggle to address the unique characteristics of LEO constellations. Single-path routing algorithms like Dijkstra's shortest path may find the minimum-latency route, but they fail to exploit the rich path diversity available in a highly connected mesh network. Furthermore, they provide no mechanism for load balancing across multiple viable paths and can potentially create bottlenecks while alternative routes remain underutilized. Recent comparative studies have shown that traditional algorithms like Dijkstra and OSPF suffer from congestion on heavily used links and increased latency under variable traffic conditions. [4] As constellation sizes grow and traffic demands increase, the need for intelligent, efficient, multi-path routing strategies becomes significant.

This research project attempts to address the fundamental challenge of optimal traffic routing in LEO satellite mega-constellations. Specifically, this research project explores an application of an Electrical flow algorithm which hopes to achieve the following objectives, minimize end-to-end latency to meet quality-of-service requirements for real-time applications, maximize network capacity utilization by intelligently distributing traffic across available paths, and provide natural load balancing to prevent congestion and improve resilience to link failures.

The problem is complicated by several factors inherent to satellite networks. First, the topology is time-varying and requires routing decisions that account for orbital dynamics. Secondly, ISLs exhibit diverse characteristics with links varying in latency (depending on distance), capacity (depending on FSO technology), and reliability. Finally, the scale of modern constellations demands computationally efficient algorithms that can operate in real-time or near-real-time.

Existing approaches present fundamental trade-offs. Shortest-path algorithms optimize for latency but ignore capacity and provide no load balancing. Max-flow algorithms optimize throughput but may produce extremely long paths. Multi-path routing protocols like OSPF with traffic engineering extensions can distribute load but require extensive manual configuration and do not naturally adapt to the unique physics of satellite networks. Recent work has explored reinforcement learning approaches for adaptive routing and pre-coded routing algorithms that exploit constellation topology regularity, yet these solutions often require significant computational overhead or lack theoretical optimality guarantees. [5] A fundamentally different approach that leverages the physical structure of the problem itself is required.

II. Background and Related Work

A. LEO Satellite Networks and Inter-Satellite Links

Low Earth Orbit satellite constellations operate at altitudes between 700 and 3,000 km, completing an orbit in less than 120 minutes. [6] This orbital velocity creates a highly dynamic network topology where satellite positions and link geometries continuously evolve. Modern LEO constellations employ inter-satellite links (ISLs) to enable direct satellite-to-satellite communication, reducing dependence on ground infrastructure and enabling global connectivity.

Free-Space Optical (FSO) communication has emerged as the dominant technology for implementing ISLs in LEO constellations. FSO links offer several advantages over traditional radio frequency systems: significantly higher data rates (up to 1 Tbps have been demonstrated experimentally [7]), immunity to radio frequency spectrum congestion, enhanced security due to narrow beam divergence, and reduced size, weight, and power requirements. [8]

However, FSO links introduce new and unique technical challenges. Pointing, acquisition, and tracking (PAT) systems must maintain precise beam alignment despite platform vibrations, orbital motion, and attitude perturbations. [8] The acquisition process can introduce delays of several seconds, and maintaining lock requires continuous fine pointing with sub-microradian accuracy. Inter-satellite link distances in LEO mega-constellations vary significantly by topology, with intra-plane links typically ranging from hundreds of kilometers to inter-plane links extending up to the 5000 km maximum range of optical laser terminals with one-way propagation delays that scales proportionally with distance from approximately 2 milliseconds for short links to 17 milliseconds at maximum range.

The network topology of LEO constellations exhibits predictable but time-varying characteristics. Constellation orbits are typically propagated using the Simplified General Perturbations 4 (SGP4) model initialized from Two-Line Element (TLE) data, though SGP4-propagated positions exhibit errors of several kilometers that grow over time. [9] The regular orbital motion allows network topology to be predicted hours or even days in advance, enabling proactive routing decisions.

B. Electrical Flow Algorithms and Graph Laplacians

Electrical flow algorithms represent a fundamentally different approach to network routing, drawing inspiration from the physics of electrical circuits. The major insight is that when electric current flows through a resistive network from a source to a sink, it automatically distributes itself across all available paths in proportion to their conductances (or inverse resistances), thereby minimizing total energy dissipation. This natural load-balancing property has an implication for network routing.

1) Theoretical Foundations

The mathematical foundation of electrical flow algorithms rests on two fundamental laws from circuit theory. Kirchhoff's Current Law states that at every node (except sources and sinks), the sum of incoming currents must equal the sum of outgoing currents—equivalent to flow conservation in network routing. Kirchhoff's Voltage Law, combined with Ohm's Law (V = IR), establishes that voltage differences drive current flow through resistances. Together, these laws give rise to a system of linear equations characterized by the graph Laplacian matrix.

For an undirected graph $G = (V, E)$ with conductances c_e on the edges, the Laplacian matrix L is defined such that $L_{ij} = -c_{ij}$ for $i \neq j$ (off-diagonal entries are negative conductances) and $L_{ii} = \Sigma_j c_{ij}$ (diagonal entries are sums of incident conductances). The electrical flow problem can then be formulated as solving the linear system $L \cdot V = b$, where V represents node voltages and b represents current injections (+1 at source, -1 at sink). Once voltages are determined, edge flows are computed using Ohm's law: $f_{ij} = c_{ij}(V_i - V_j)$.

The major benefit of this formulation is in its connection to optimization. The electrical flow between two nodes minimizes the energy dissipation $\Sigma_e\, r_e\, f_e^2$ subject to flow conservation constraints, where $r_e = \frac{1}{c_e}$ is the resistance of edge e. This variational characterization ensures that electrical flows naturally find the "best" distribution of traffic across available paths without requiring explicit multi-path computation or load-balancing heuristics.

2) Computational Efficiency

A major breakthrough in the practical application of electrical flow algorithms came from advances in fast Laplacian system solvers. [10] introduced an approach to computing approximately maximum flow in undirected graphs by solving a sequence of electrical flow problems, with each electrical flow given by the solution of a Laplacian system that can be approximately computed in nearly-linear time. Their algorithm computes a $(1 + \varepsilon)$ approximately maximum s-t flow in time $\tilde{O}(m^{\left(\frac{4}{3}\right)}\varepsilon^{-\frac{11}{3}})$ for graphs with n vertices and m edges.

This work built upon concurrent advances in fast solvers for symmetric diagonally dominant (SDD) matrices, of which graph Laplacians are a special case. [12] and Spielman and Teng [11] demonstrated near-linear time algorithms for solving these systems. Using these solvers, a vector of vertex potentials and an s-t flow can be computed in time $\tilde{O}(m\ log\ n\ log(\frac{1}{\varepsilon}))$ for any desired accuracy ε. These theoretical advances have been complemented by practical implementations that leverage iterative methods such as preconditioned conjugate gradient, making electrical flow computation tractable even for large-scale networks.

The computational efficiency of electrical flow algorithms becomes particularly attractive for LEO satellite networks where the graph is sparse (each satellite connects to only a handful of neighbors) and the topology, while time-varying, changes predictably. The combination of sparse structure and fast solvers enables near-real-time routing computation even for constellations with thousands of satellites.

3) Applications to Network Routing

While electrical flows were initially studied in the context of theoretical algorithm design, recent work has explored their application to practical routing problems. Robust routing using electrical flows has been applied to road network navigation, demonstrating that electrical flow-based routing provides natural load balancing and robustness to network changes. In this context, edge resistances are set based on travel time, and the resulting electrical flows identify multiple alternative routes with appropriate traffic distribution.

However, electrical flow algorithms have not been previously applied to satellite network routing. The space domain presents unique opportunities and challenges: the predictable topology enables offline computation and proactive route planning, but the high-speed motion and limited on-board computational resources require efficient algorithms. The combination of multiple optimization objectives (latency, capacity, reliability) naturally maps to the resistance formulation, where edge resistances can capture trade-offs between these competing factors. Moreover, the inherent multi-path nature of electrical flows aligns well with the path diversity available in densely connected LEO mesh networks.

III. Methodology

This section details the electrical flow routing approach for LEO satellite constellations. The section begin by establishing the mathematical framework that maps network routing to electrical circuit theory, then describe the algorithmic implementation and constellation simulation infrastructure.

A. Circuit Analogy Framework

The foundation of the approach rests on a mapping between satellite network elements and electrical circuit components.

1) Network-to-Circuit Mapping

The satellite constellation can be modeled as an undirected graph $G = (V, E)$, where V represents the set of satellites (nodes) and E represents the set of inter-satellite links (edges). Each element in the network maps to an electrical equivalent as shown in Table 1.

Table 1: Network-to-Circuit Element Mapping

Network Element	Electrical Analog	Interpretation
Satellite Node	Junction/Node	Point where currents meet
Inter-Satellite Link (ISL)	Resistor	Element that impedes flow
Data Traffic Demand	Electric Current	Flow Quantity
Source Satellite	Positive Voltage Terminal	Current Injection Point
Destination Satellite	Ground (0V)	Current extraction Point

Link Latency	Resistance Component	Flow Impedance
Link Capacity	Conductance Component	Flow Facilitation

This mapping enables us to reformulate the routing problem as a circuit analysis problem. The data traffic flow from source to destination becomes analogous to electric current flowing from a voltage source to ground, with the network topology determining how current distributes across parallel paths according to fundamental circuit laws.

2) Resistance Model

The critical component of the circuit analogy is the resistance assignment function, which encodes link characteristics into electrical resistances. For each inter-satellite link $(u,v) \in E$, we define the resistance as $R(u,v) = \alpha \cdot \left(\frac{latency(u,v)}{\lambda_{norm}}\right) + \beta \cdot \left(\frac{\kappa_{norm}}{capacity(u,v)}\right)$ where $latency(u,v)$ is the propagation delay in milliseconds, $capacity(u,v)$ is the link bandwidth in Gbps, α and β are tunable weighting parameters, and λ_{norm} and κ_{norm} are normalization constants. The weighting parameters α and β control the optimization objective. Setting $\alpha > \beta$ prioritizes low-latency paths at the expense of capacity, suitable for real-time applications. Conversely, $\beta > \alpha$ maximizes throughput for bulk data transfer. Equal weights $(\alpha = \beta)$ balance both objectives. The normalization constants ensure numerical stability when solving the resulting linear system. In the implementation for this final project, $\lambda\_norm = 100$ ms (typical LEO ISL latency range) and $\kappa\_norm = 10$ Gbps (representative optical link capacity).

B. Mathematical Formulation

The electrical flow routing algorithm solves for the voltage distribution across the satellite network, from which edge flows are derived using Ohm's law. This formulation naturally incorporates flow conservation and optimal load distribution.

1) Kirchhoff's Current Law (Flow Conservation)

Kirchhoff's Current Law states that for any node in an electrical circuit, the sum of currents flowing into that node equals the sum of currents flowing out. In our network context, this translates to flow conservation at each satellite: $\sum(flow\ into\ node\ i) = \sum(flow\ out\ of\ node\ i)$ for all $i \in V \setminus \{s,d\}$, where s is the source satellite and d is the destination satellite. This constraint ensures that intermediate satellites neither generate nor consume traffic and that they purely relay data. Mathematically, for each node i, we require: $\sum\_\{j \in N(i)\} I_{\{ji\}} = \sum\_\{k \in N(i)\} I_{\{ik\}}$, where $N(i)$ represents the neighbors of node i, and $I_{\{ji\}}$ represents current (flow) from node j to node i.

2. Ohm's Law (Voltage-Flow Relationship)

Ohm's Law relates voltage difference to current through resistance. For each link (u, v) in the network, the flow is determined by: $I(u, v) = (V_u - V_v) / R(u, v) = G(u, v) \cdot (V_u - V_v)$ where V_u and V_v are the voltage potentials at nodes u and v, $R(u, v)$ is the link resistance, $G(u, v) = \frac{1}{R(u,v)}$ is the conductance, and $I(u, v)$ is the resulting current (traffic flow). This relationship captures the intuitive notion that traffic preferentially flows through low-resistance (high-quality) links, with the flow magnitude proportional to the voltage difference.

3. Graph Laplacian System

The Laplacian matrix, also called the graph Laplacian or Kirchhoff matrix, is a matrix representation of a graph that combines degree and adjacency information. By combining Kirchhoff's Current Law with Ohm's Law across all nodes, we obtain a system of linear equations expressed in terms of the graph Laplacian matrix L: $L \cdot V = b$ where $L \in \mathbb{R}^{n \times n}$ is the Laplacian matrix, $V \in \mathbb{R}^n$ is the vector of node voltages (unknowns), and $b \in \mathbb{R}^n$ is the current injection vector. The Laplacian matrix is defined as $L = D - A$, where D is the degree matrix and A is the adjacency matrix weighted by conductances. Specifically:

$$L_{\{ij\}} = \begin{cases} \sum_{\{k \in N(i)\}} G(i, k) \;\; if\ i = j \\ -G(i, j) if\ (i, j) \in E \\ 0\ otherwise \end{cases}$$

The current injection vector b encodes the traffic demand:

$$b_i = \begin{cases} +F\ if\ i = source \\ -F\ if\ i = destination \\ 0\ otherwise \end{cases}$$

where F is the traffic demand in units of flow (in Gbps).

4. Solution Method

The Laplacian matrix L is singular because it has a zero eigenvalue corresponding to the uniform voltage vector (all nodes at the same potential). To obtain a unique solution, we fix the destination node voltage to zero (ground reference) $V_d = 0$. This yields a reduced system $L_{reduced} \cdot V_{reduced} = b_{reduced}$, where we remove the row and column corresponding to the destination node. The Laplacian matrix is sometimes referred to as the nodal admittance matrix in electrical engineering and is necessary for circuit analysis. The reduced system is solved using sparse linear algebra techniques. In the final project's implementation, we employ the spsolve function from SciPy's sparse linear algebra library, which uses either direct methods (LU decomposition for small systems) or iterative methods (conjugate gradient) depending on the system size and sparsity pattern. Once node voltages V are computed, we calculate the flow on each edge (u, v) using $Flow(u,v) = G(u,v) \cdot (V_u - V_v)$. Positive flow indicates traffic moving from u to v, while negative flow indicates the reverse direction.

5. Path Extraction

After computing edge flows, we extract the actual routing paths using a flow-following algorithm. Starting from the source node, we iteratively select the outgoing edge with the largest positive flow, adding it to the path and moving to the next node. This process continues until reaching the destination. The algorithm naturally discovers multiple paths by considering edges with significant flow values above a threshold (e.g., 10% of maximum flow). This threshold filters out negligible flows that result from numerical precision limitations while capturing all meaningful parallel paths.

C. LEO Constellation Simulation Infrastructure

To validate our electrical flow routing algorithm with realistic satellite network conditions, we developed a comprehensive LEO constellation simulation environment that incorporates orbital mechanics, link budget calculations, and dynamic topology generation.

1. Orbital Propagation

We use the Simplified General Perturbations 4 (SGP4) model for satellite orbital propagation, which is used for satellites close enough to Earth that their orbit takes less than 225 minutes to complete. The SGP4 propagator operates on Two-Line Element (TLE) data obtained from public sources. TLE data provides mean orbital elements at a specific epoch, including semi-major axis (or mean motion), eccentricity, inclination, right ascension of ascending node, argument of perigee and mean anomaly. The SGP4 model has a typical error of approximately 1 km at epoch and grows at 1-3 km per day, which is sufficient accuracy for network topology analysis where inter-satellite link ranges span hundreds to thousands of kilometers.

Our implementation downloads real-time TLE data for operational satellites (primarily Starlink constellation) from CelesTrak, a public repository maintained by the Center for Space Standards and Innovation. The SGP4 propagator computes satellite positions at any desired time by accounting for Earth's oblateness (J2 perturbations), atmospheric drag effects, and solar/lunar gravitational perturbations (for higher orbits). The propagator outputs satellite positions in the True Equator Mean Equinox (TEME) reference frame, an

Earth-centered inertial coordinate system suitable for computing inter-satellite distances and link geometries.

2. Free-Space Optical Link Budget

Inter-satellite links in modern LEO constellations predominantly use Free-Space Optical (FSO) communication due to its high bandwidth, low power consumption, and immunity to radio frequency interference. We model FSO link viability using a simplified link budget calculation $P_{rx} = P_{tx} + G_{tx} + G_{rx} - L_{space} - L_{pointing} - L_{atm}$ where P_{rx} = received optical power (dBm), P_{tx} = transmitted optical power (dBm), G_{tx}, G_{rx} = transmit and receive antenna gains (dBi), L_{space} = free-space path loss (dB), $L_{pointing}$ = pointing loss due to beam divergence (dB), and L_{atm} = atmospheric attenuation (negligible for vacuum ISLs).

The free-space path loss for optical wavelengths (typically 1550 nm) is $L_{space} = 20 \cdot log_{10}(4\pi \cdot d/\lambda)$ where d is the link distance in meters and λ is the optical wavelength. A link is considered viable if P_{rx} exceeds the receiver sensitivity threshold (typically -40 to -35 dBm for coherent optical receivers). We also enforce geometric constraints with maximum link distance: 5000 km (typical for inter-plane links), and minimum elevation angle: 25° (to avoid Earth occultation).

3. Network Topology Construction

For each simulation time step, we construct the network topology $G(t) = (V, E(t))$ where the edge set $E(t)$ varies with satellite positions.

Algorithm: Dynamic Topology Generation

Input: Satellite positions $\{p_1, p_2, ..., p_{\mathrm{n}}\}$, time t

Output: Network graph $G(t)$

1. Initialize empty graph $G = (V, \emptyset)$

2. For each pair of satellites (i, j):

  a. Compute distance $d\_ij = ||p\_i - p\_j||$

  b. If d_ij ≤ max_distance:

    i. Compute link budget parameters

    ii. Calculate latency = d_ij / c

    iii. Estimate capacity from SNR

iv. If link viable:

- Add edge (i, j) to G and assign resistance R_ij

3. Return G(t)

The speed of light in vacuum c = 299,792 km/s yields latencies ranging from 3.3 ms per 1000 km of link distance. Link capacity is estimated from the signal-to-noise ratio (SNR) derived from the link budget using Shannon's capacity formula $Capacity = Bandwidth \cdot log_2(1 + SNR)$. Typical FSO systems achieve 10-100 Gbps per link depending on modulation format, bandwidth, and link margin.

4. Implementation Parameters

The implementation uses the following parameters based on typical LEO constellation characteristics. For Constellation Parameters: Altitude= 550 km (Starlink shell), Inclination= 53° (mid-latitude coverage), Number of satellites= 50-1000 (scalability testing), and Orbital period= ~95 minutes. For Link Parameters: Maximum ISL distance= 5000 km, FSO wavelength= 1550 nm (C-band), Transmit power= +20 dBm, Receiver sensitivity= -40 dBm, and Minimum elevation: 25°. Finally, for the Resistance Model Parameters: α (latency weight)= 1.0, β (capacity weight)= 1.0, Latency normalization= 100 ms, and Capacity normalization= 10 Gbps. These parameters can be adjusted to explore different constellation architectures and optimization objectives.

IV. Results and Analysis

The electrical flow routing algorithm on a simulated 100-satellite LEO constellation with realistic orbital mechanics and FSO link parameters. The network topology consisted of 100 nodes with 1,426 bidirectional inter-satellite links (average degree 28.52, diameter 6 hops), representing a densely connected mesh network characteristic of modern mega-constellations.

A. Routing Performance Comparison

Table 2 presents the comparative performance of electrical flow routing against three baseline algorithms: Dijkstra's shortest path, OSPF bandwidth-aware routing, and ECMP multi-path routing, for a randomly selected source-destination pair (nodes 0 → 99).

Algorithm	Paths	Latency (ms)	Capacity (Gbps)	Compute Time (ms)
Electrical Flow	5	62.64	10.00	14.068
Dijkstra	1	61.04	10.00	0.999
OSPF	1	62.64	10.00	19.295
ECMP	1	61.04	10.00	3.522

The electrical flow algorithm discovered 5 parallel routing paths compared to single paths for baseline algorithms. End-to-end latency was 62.64 ms, representing a 2.6% increase over Dijkstra's optimal 61.04 ms. Computation time of 14.07 ms was faster than OSPF (19.30 ms) but slower than Dijkstra (1.00 ms), remaining well within acceptable bounds for near-real-time routing in predictable LEO topologies.

B. Load Balancing Analysis

The electrical flow approach demonstrated superior load distribution characteristics. With 568.69 effective paths (derived from the Herfindahl-Hirschman Index), the algorithm utilized all 1,426 available network edges, compared to the single-path utilization of baseline methods. The Gini coefficient of 0.526 indicated moderate flow concentration. Which is not perfectly uniform, but substantially better distributed than any single-path approach. Normalized entropy of 0.930 (on a 0-1 scale) confirmed that flow was well-dispersed across the network topology. The max/avg flow ratio of 7.95 shows that while some edges carried higher loads, no single link was overwhelmingly congested.

These metrics validate that electrical flow routing naturally exploits the path diversity inherent in dense LEO mesh networks. By distributing traffic across hundreds of effective paths, the algorithm provides inherent resilience to link failures and congestion hotspots which are critical properties for operational satellite constellations where individual links may fail due to pointing errors, hardware faults, or interference.

C. Computational Scalability

The near-linear time complexity of sparse Laplacian solvers suggests favorable scaling properties for larger constellations. For the 100-satellite test case, solving the 100×100 system required only 14 ms on standard computational hardware. Since the Laplacian matrix sparsity is determined by local network connectivity rather than total satellite count, the system remains sparse as constellation size grows. This sparse structure enables efficient solution using iterative methods, with computational requirements scaling approximately linearly with the number of satellites for fixed average connectivity.

V. Conclusion and Future Work

This research project demonstrates that electrical flow algorithms offer a promising approach to traffic routing in LEO satellite mega-constellations. By modeling the network as an electrical circuit and solving the graph Laplacian system, the method achieves multi-path load balancing with minimal latency penalty compared to traditional shortest-path routing. The algorithm naturally distributes traffic across hundreds of effective paths, providing resilience to link failures and congestion. Key findings include: (1) electrical flow routing discovers significantly more parallel paths than baseline algorithms while maintaining comparable latency, (2) load distribution metrics indicate effective utilization of network path diversity, and (3) computation times remain tractable even for large constellation scales due to sparse matrix structure. However, more research and experiments should be conducted, especially for a more comprehensive LEO satellite architecture, explore increased complexity in resistance modeling (time-varying weights that account for link reliability, thermal constraints, and QoS requirements), explore implementation details, run a longer simulation over a simulated time frame of 24 hours, in addition to comparing to more recent algorithm proposals in LEO constellations (including to recreating research in RL algorithms and the SDN approach), and find a realistic network flow data set to incorporate into the simulation or conduct statisticial network flow generation.

The codebase for the simple model can be found here:
https://github.com/nickyduesxd/ElectroFlow.git

VI. References

[1] Statista, “Active satellites by operator 2025,” *Statista*, Jul. 2025. [Online]. Available: https://www.statista.com/statistics/1224643/active-satellite-by-operator/

[2] Starlink, “Technology,” Starlink, 2025. [Online]. Available: https://starlink.com/technology

[3] A. A. Ahmed, “A Comprehensive Review of Satellite Orbital Placement and Coverage Optimization for Low Earth Orbit Satellite Networks: Challenges and Solutions,” *Network*, vol. 5, no. 3, p. 32, Aug. 2025. [Online]. Available: https://doi.org/10.3390/network5030032

[4] Z. Feng, B. Li, H. Ding, and F. Hou, “An integrated routing and data fragmentation strategy for optimizing end-to-end delay in LEO satellite networks,” Ad Hoc Networks, vol. 175, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1570870525001180

[5] Y. Shi, W. Wang, X. Zhu, and H. Zhu, “Low Earth Orbit Satellite Network Routing Algorithm Based on Graph Neural Networks and Deep Q-Network,” *Applied Sciences*, vol. 14, no. 9, article 3840, 2024. [Online]. Available: https://doi.org/10.3390/app14093840

[6] *Low Earth Orbit*, ScienceDirect Topics, “Low Earth orbit,” [cited 2025 Dec. 01]. Available: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/low-earth-orbit

[7] F. Moll, M. Brehm, A. Schwarz, M. Gschossmann, C. M. Schieler, M. Grözing, J. Roths, M. Berroth, and J. Pina, "Tbit/s line-rate satellite feeder links enabled by coherent modulation and full-adaptive optics," Light: Science & Applications, vol. 12, no. 1, p. 144, Jun. 2023. Available: https://www.nature.com/articles/s41377-023-01201-7

[8] NASA, "Communications," NASA Small Spacecraft Systems Virtual Institute, Feb. 2025. [Online]. Available: https://www.nasa.gov/smallsat-institute/sst-soa/soa-communications/

[9] D. Vallado and P. Crawford, "SGP4 orbit determination," in Proc. AIAA/AAS Astrodynamics Specialist Conf. and Exhibit, Honolulu, HI, Aug. 2008, Paper 6770.

[10] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S.-H. Teng, "Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs," in Proc. 43rd Annual ACM Symp. on Theory of Computing (STOC), San Jose, CA, Jun. 2011, pp. 273–282.

[11] D. A. Spielman and S.-H. Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," SIAM J. Computing, vol. 43, no. 2, pp. 1–88, 2014.

[12] I. Koutis, G. L. Miller, and R. Peng, "A nearly-m log n time solver for SDD linear systems," in Proc. 52nd Annual IEEE Symp. on Foundations of Computer Science, Palm Springs, CA, Oct. 2011, pp. 590–598.