Seam-Aware Location-Based Random Walk Routing Algorithms for Low Orbit Satellite Constellations

Oren Markovitz and Michael Segal
School of Electrical and Computer Engineering
Ben-Gurion University, Beer-Sheva, Israel

***Abstract*—As of 2018, several low orbit (LEO) constellations are being designed and planned. These include SpaceX, OneWeb, LeoSat, Telesat and others. Some of these constellations include Inter-Satellite Links (ISL) communication at the initial or second phase as well as on-board processing capabilities. The LEO constellations create a network that includes the satellites (as routing nodes) connected by ISLs, and the satellite terminals that dynamically connect to one of the satellites. The LEO network presents unique challenges to traffic routing and service planning due to dynamic changes in the network topology (interconnection between satellites, and between satellites and terminals). In addition, the LEO latency (which is low, compared to GEO and MEO) is significant when using legacy routing protocols (each ISL latency can be in the order of 10 mSecs or more and ground to satellite latency is in the order of 10 mSecs). In case of a polar constellation, the LEO satellite orbit is south-to-north on one half of the constellation and north-to-south on the other half. As a result, there are neighboring planes in which satellites are moving in opposite directions. Satellites can easily establish and maintain ISLs with neighboring satellites on the same plane. However, a link with a neighboring satellite on the adjacent plane can only be established if the satellite on that plane is moving in the same direction. The barriers between the two satellite groups are called seams. This paper is the first to analyze the impact of the seam on location based routing in a polar constellation. We propose an asymmetric seam-aware location-based routing algorithm, and use a random walk on a geographical shortest path lattice for load balancing.**

***Index Terms*—LEO Routing, SEAM, LEO SLA, AGR, SAGRW, Random Walk.**

I. Introduction

The LEO constellations create a network that includes the satellites (as routing nodes) connected by ISLs, and the satellite terminals that dynamically connect to one of the satellites. The satellite grid movements vs. the earth and the ground nodes (terminals) force the terminals to switch beams and/or satellites. The switch of beams and satellites (aka handover) requires a change in the routing information of the satellites. The rate of changes may be very high, pending the geographical distribution of the terminals, while the latency to propagate these changes to all the satellites is in the order of 100 mSecs. These changes are frequent (a LEO terminal may change satellite every 10 min) with possible transients of high rates (for densely populated service areas). The combination of transient high rate changes with high latency presents a unique challenge for designing a routing protocol that can support the frequent changes without packet drops.

This work addresses the problem of location-based routing from a source terminal to a destination terminal in a polar constellation, while considering the seam barrier. Current location-based algorithms offer low complexity (compared to existing LEO routing algorithms), but do not account for access routing and do not address routing across the seam barrier or the total induced load. We present a novel location-based routing algorithm - Seam-Aware Asymmetric Geographical Random Walk Routing (SAGRW).

The paper is organized as follows. Section II reviews the main offering of each approach. We present in Sections III and IV an innovative location routing protocol with reduced complexity and support for access routing and for polar constellations cross-seam routing. Section V shows some experimental results for SAGRW. Section VI summarizes the benefits of the SAGRW algorithm introduced in this work.

II. Related Work

Related work on LEO networking spans over a wide range of research areas including: hybrid GEO and LEO networking [12], optimizing the network throughput by changing the transmission time, optimizing the satellite receive and transmit queues and QoS [10]. LEO routing related work can be split into four main paradigms – virtual topology routing [1], [8], [9], [11], [13], [17], [18], virtual node routing [3]–[7], location-based routing [2], [11] and demand island routing [16].

Virtual topology routing models the dynamic nature of the network as a series of static fixed configurations or snapshots. In each snapshot the satellite topology, terminals and GW connections are fixed. For each snapshot it is possible to perform routing or circuit planning using legacy networking algorithms. The time frame for each snapshot is determined by the algorithm. Legacy routing algorithms are used to optimize the routing for each static configuration. In [17], [18], the generated snapshot takes into account the seam impact on the availability of ISLs. The virtual topology approach has inherent shortcomings: dynamic real-time functions such as QoS, and link failures events are not handled by virtual topology routing. The computational complexity (number of snapshots) is high. An exact time synchronization for switching snapshots is required for all terminals, satellites and GWs.

In the virtual node approach, the constellation of satellites is replaced by a virtual constellation with fixed locations for the virtual satellites. At a given time, each physical satellite

978-1-6654-2854-5/21/$31.00 ©2021 IEEE

is assigned to a virtual satellite. As the satellite constellation is moving, the coupling changes according to the identity of the satellite closest to the virtual spot. Each virtual satellite has a routing table indicating the next hop for any destination satellite. When a physical satellite is assigned to a virtual node, it will use the virtual node routing table until it is re-assigned to a different virtual node due to constellation movement. The virtual node algorithms do not specify how the destination satellite identity is determined based on the destination terminal address. A key assumption for the virtual node algorithms is that since the speed of satellite movement is far smaller than the information transmission, the satellite topology is stationary during a packet's life-span.

In the Demand Island paradigm, the demand is categorized by its geographical properties and is split into autonomous Demand Islands. The network architecture is one-to-many. In addition, the geographical area assigned to each island is a rectangle. The combined properties of a rectangle graph with a single serving GW give way to a low complexity service planning algorithm that utilizes the full GW capacity and supports real time routing for terminals that are connected to multiple satellites.

Location-based algorithms use the terminals' geographical location as an additional attribute to calculate the distances between the terminals and the current satellite for each next hop satellite. Each routing step tries to minimize the physical distance to the destination terminal. Current location-based algorithms do not determine how the last satellite on the path will route the packet received to the proper user beam. Location-based routing does not enable load balancing of traffic (all traffic to a given geographical location will flow on the same path) and does not allow service planning, since mapping of traffic on top of the grid is dynamic. Moreover, the algorithms do not address the problem of routing traffic across the seam (in case of polar constellation) which may prevent direct routing between two points for a few hours each day. Privacy may present yet another challenge as in many cases customers (and even legislation) may prevent disclosing the exact geographical location of the terminal (customer). Some of the problems that are addressed by random walk algorithms include stochastic process analysis [14] and shortest path with local knowledge [15].

III. Asymmetric Geographical Routing

We present Asymmetric Geographical Routing (AGR) that is an extension of the location-based algorithms [2]. In location-based algorithms, the satellite calculates the next hop based on its location and the destination terminal location. The algorithm does not handle overlapping of satellites and handover transitions, both of which may result in miss-routing of the final hop. To overcome these shortcomings, current work suggests that each satellite will update its neighbours with the list of terminals connected to it. This requirement significantly increases the satellite payload memory size.

AGR provides low complexity access routing by assigning each terminal to a fixed GW. Future load balancing process may be used to assign the terminal to different/multiple GWs. For the sake of routing and service planning, the assignment is fixed. In current wire-line and satellite routing approaches, the addressing scheme of the terminals is static and symmetric. The addressing scheme used by AGR is asymmetric and partially dynamic. The GW address is static and is a function of its geographical location. The terminal address is composed of static (unique ID) and dynamic components. The dynamic component includes the satellite ID and user beam number on which the terminal is logged-on. Packets from the terminal to the GW are routed using geographical routing. The number of GWs is significantly smaller than the number of terminals, and the access routing of the last node can be resolved by local routing as described in [2] but at a much lower cost.

Packets from the GW to the terminal can be sent once a packet is received from the terminal. This should not be an issue as most sessions are initiated by the terminal (i.e. client-server model). Once a packet is received from the terminal, the GW will register it and keep track of each terminal satellite ID and user beam. Packets sent from the GW to the terminal can be routed using any of the virtual node algorithms.

To account for handovers, AGR complements these algorithms by resolving the destination satellite ID and the beam ID (not handled by current work). In addition, AGR is automatically updated to support terminal handover between satellites. When the terminal establishes a connection with a rising satellite it will update its source address accordingly.

AGR handles access (satellite to terminal) routing, handover, and identification of the destination satellite, all of which are not handled by current virtual node and location-based routing algorithms. Furthermore, it simplifies the access routing of the geographical routing and does not require end-customers to share their geographical location.

IV. Seam-Aware Geographical Routing

The following subsections present the seam disconnection problem, model the seam impact on latency and routing, and calculate the cycle of impact as a function of the two end points locations. Finally, a novel solution for shortest path geographical routing is presented with a pseudo code of the full SAGRW routing algorithm.

A. Seam Disconnections

In case of a polar constellation, the satellite grid is split into two disconnected planes (Figure 1). The barriers between the two planes are called seams.

The seam is modeled using a graph (Figure 2), in which the two sub-grids are concatenated. The satellite planes form a cylinder with seams at both ends. The satellites on each plane are connected and adjacent planes within the cylinder are connected to each other.

B. Analysis of Seam Impact on Service Latency

Figure 3 demonstrates how a connection between two terminals is impacted by the seam shift: E_1 and E_2 are connected using the satellite grid. The satellite grid horizontal movement

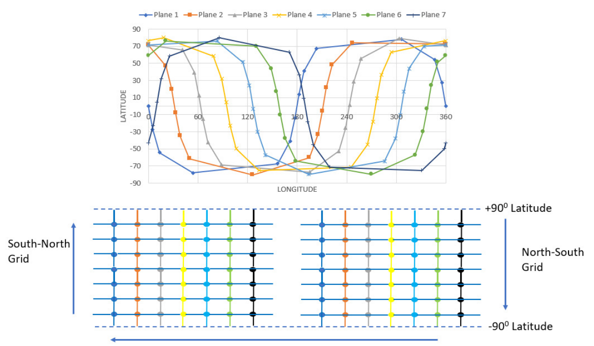

Fig. 1: LEO Constellation Seam Satellite Grids.

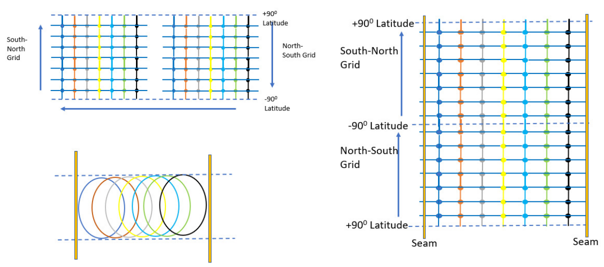

Fig. 2: Seam Modeling.

causes E_2 to connect to a satellite on the north-south grid while E_1 is located on the south-north grid. The shortest path between the two end points is changed (marked by a dashed line). Using the seam modeling, it is possible to calculate an upper bound on the latency change for any given connection, caused by the seam shifts.

Let:

- E_1, E_2 be two endpoints (an endpoint is a terminal or a GW) communicating over the satellite grid.
- $S_{1,1} \ldots S_{m,n}$ and $S'_{1,1} \ldots S'_{m,n}$ be the satellites on the south-north and north-south grids respectively.
- $S_{i,j}$ be connected to its adjacent satellite on the same plane $S_{i,j+1}$, and $S_{i,1}$ be connected with $S_{i,n}$.
- $S_{i,j}$ be connected to the satellite on its adjacent plane $S_{i+1,j}$.
- $S_{i,n}$ be connected to its adjacent satellite on the same plane $S'_{i,n}$, and $S'_{i,1}$ be connected with $S_{i,1}$.

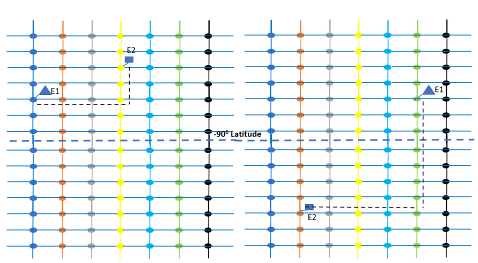

Fig. 3: Seam Shift Impact on the Shortest Path Between Two end points.

- $S'_{i,j}$ be connected to the satellite on its adjacent plane $S'_{i+1,j}$.
- E_1 be connected to S_{x_1,y_1} and E_2 to S_{x_2,y_2} on the same satellite sub-grid.
- The seam shifts horizontally, causing the endpoints to move by $d \leq m$ planes, and thus, after the seam shift E_1, E_2 are located on different grids.

Lemma 1. *The length of the shortest path (SP) between E_1, E_2 measured in hops is $|x_2 - x_1| + |y_2 - y_1|$ when located on the same grid. The length of the shortest path between E_1, E_2 after the seam shift (causing E_1, E_2 to be located on different grids), is* $\min(|x_2 - x_1 - m|, |x_1 - x_2 - m|) + \min(y_1 + y_2), 2n - (y_1 + y_2))$.

Proof. It is easy to see that the length of the shortest path between E_1, E_2 is $|x_2 - x_1| + |y_2 - y_1|$ when located on the same sub-grid.

When E_1, E_2 shift by $d \leq m$ horizontal links then:

- E_1 is handed over from S_{x_1,y_1} to S_{x_1+d,y_1}.
- E_2 is handed over from S_{x_2,y_2} to $S'_{(x_2+d) \mod m, y_2}$ (the sub-grid mirrors the sub-grids on the $-90°$ latitude line - Figure 3).

The new shortest path length is the horizontal length of $x_1 + d -$ $(x_2 + d + m)$ and vertical length of $\min((y_1 + y_2), 2n - (y_1 + y_2))$.

$$SP = \min(|x_2 - x_1 - m|, |x_1 - x_2 - m|) + \min((y_1 + y_2), (2n - (y_1 + y_2))$$

□

When the two end points are separated, the latency is increased when the horizontal distance on the same grid is smaller.

C. Seam Cycle

This section provides further analysis of the seam impact by calculating the cycle of the impact. The satellite grid horizontal speed is equal to the earth spin speed. The seam completes a full orbit of the earth every 12 hours (seam cycle). For every connection the seam cycle has different properties. The cycle is split into two parts, a short-latency period (SLP), in which the two end points are located on the same satellite grid, and a long-latency period (LLP), in which the two end points communication crosses the seam. The SLP is calculated as a function of the horizontal physical distance between the two end points. It is easy to see that $SLP = 12\ (x_1 - x_2)/EP$ where E_1, E_2 physical satellite locations are (x_1, y_1) and (x_2, y_2) respectively and EP is half of the earth perimeter. Alternatively, the SLP can be calculated as a function of the horizontal grid index distance between the two end points.

D. Handling the Seam Impact on Location-Based Routing Algorithms

The seam impact on the SLA, and the SLA cycle varies as a function of the two end points physical location. When one of the end points is handed over to a rising satellite, a new route is calculated. In case of seam impact, the new route latency and bandwidth may significantly change.

In case of location-based algorithms, the next hop is calculated by the satellite such that the distance to the destination end point is minimized. We propose a Seam-Aware Geographical Routing algorithm (SAGRW) to support dynamic route across the seam.

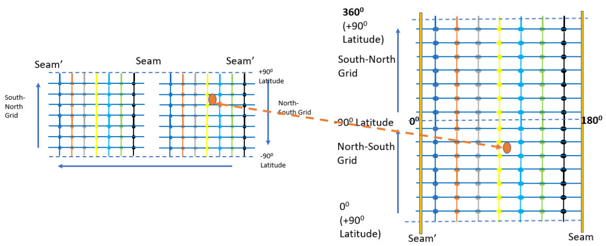

Fig. 4: Mapping GEO Coordinates to Seam Model.

In SAGRW, the geographical locations of the end points and satellites are mapped to the seam model (Figure 4). Once the coordinates are mapped, the location-based algorithm can be executed. The following details the mapping routine:

Let:

- $SEAM$ be the horizontal coordinate of the seam.
- x, y be the horizontal and vertical coordinates of the node we want to map.
- x', y' be the mapped horizontal and vertical coordinates.

Mapping $(x, y) \to (x', y')$:

- $SEAM' = (SEAM + 180°) \mod 360°$.
- $x' = (x - SEAM' + 360°) \mod 180° - SEAM$.
- $SEAM \le x < SEAM' \implies y' = 90° - y$, otherwise $y' = y + 270°$.

E. SAGRW Algorithm

SAGRW calculates the seam location and maps the satellite location and the destination location for each received packet. Once the destination and satellites are mapped, SAGRW executes the location-based routing on the mapped coordinates such that each satellite sends the traffic to the next satellite that shortens the distance to the destination. In each step, the algorithm advances either horizontally or vertically toward the destination. The algorithm generates a lattice of shortest paths between the sender and the receiver. A load based normal distribution random walks is used to optimize the load distribution on the lattice. The random walk randomly selects a horizontal or vertical step using a normal distribution random function.

Let:

- $S_{1,1} \dots S_{m,n}$ be the constellation satellites.
- $e_{(i,j)(v,w)}$ be an edge (ISL) connecting satellite $S_{i,j}$ to $S_{v,w}$.
- $term$ be a terminal connected to the satellite grid through satellite S user beam B.
- $G_{x,y}$ be a GW serving $term$ located at coordinates x, y.
- $SEAM$ be the seam horizontal location.
- $Random(0, 1)$ be a normal distribution random function over $\{0, 1\}$ returning 0 or 1 at equal probability.

Algorithm 1 - SAGRW Algorithm

Terminal routine, Input: $term$ connected to satellite S user beam B, sends a traffic demand d to $G_{x,y}$.

1) Add a label to each packet (x, y, S, B).
2) Send the packets to the satellite serving the terminal.

Satellite routine, Input: $S_{i,j}$ located at coordinates (x_s, y_s) receives traffic d from the user beams and ISLs.

1) Extract the destination coordinates (x,y) from the demand.
2) $SEAM' = (SEAM + 180°) \mod 360°$.
3) $x' = (x - SEAM' + 360°) \mod 180° - SEAM$.
4) if $SEAM \le x < SEAM'$ then $y' = 90° - y$, otherwise $y' = y + 270°$.
5) If $G_{x,y}$ is connected to this satellite, send d to the feeder link.
6) If $G_{x,y}$ is connected to $S_{i,j}$ via edge e, send d on e.
7) $H = V = None$.
8) If $x' < x_s$ then $H = e_{(i,j)(i+1,j)}$.
9) If $x' > x_s$ then $H = e_{(i,j)(i-1,j)}$.
10) If $y' < y_s$ then $V = e_{(i,j)(i,j+1)}$.
11) If $y' > y_s$ then $V = e_{(i,j)(i,j-1)}$.
12) if $V = none$ or $Random(0, 1) = 1$, send d on H. else, send d on V.

GW routine, Input: traffic d from the satellite feeder link.

1) Extract the terminal satellite S and user beam B from the demand.
2) Store $(term, S, B)$.
3) Process demand.

Lemma 2. *When running the SAGWR algorithm, the induced load of sending* d *demand capacity from a source terminal to a destination terminal on edge* $e_{(i,j)(u,v)}$ *is* $\dfrac{d\binom{i+j}{j}}{2\sum_{n=1}^{j}\binom{i+j}{n}}$

Let:

- $S_{1,1}$ be the satellite serving the source terminal.
- $S_{w,z}$ be the satellite serving the destination terminal.
- d be the demand capacity sent from the source terminal to the destination terminal using the SAGWR algorithm.
- $S_{1,1} \dots S_{w,z}$ be the set of satellites in the geographical shortest path lattice of satellites used by the SAGWR algorithm.
- $e_{(i,j)(u,v)}$ be an edge (ISL) connecting satellite $S_{i,j}$ to $S_{u,v}$.

Proof. First, we map the geographical shortest path lattice on a Pascal triangle such that the sender terminal satellite $S_{1,1}$ is located at the top vertex of the triangle. The number of lines in the triangle is $\max\{w, z\}$ as detailed in Figure 5. A node

$S_{i,j}$ on the geographical shortest path lattice is mapped to line $i+j$ with index j.

In Pascal triangle the total number of paths received on a node with index k at line n is $\binom{n}{k}$. Since we use a normal distribution with equal probability, then the number of paths is equal to the number of packets received on a node (assuming each packet traverses a different path). The total number of paths from the source of length $i+j$ is therefore $\sum_{n=1}^{j}\binom{i+j}{n}$. We calculate the total number of paths at line $i+j$ which is $\sum_{n=1}^{j}\binom{i+j}{n}$. This is equivalent to the total number of packets sent from the source node. The fraction of the traffic received by a node $S_{i,j}$ is $\dfrac{d\binom{i+j}{j}}{2\sum_{n=1}^{j}\binom{i+j}{n}}$. Since we use normal distribution of 0.5, the traffic is evenly split by the random walk between the two outgoing edges. When the node is located on the boundary of the lattice (i.e. j=z) the traffic is not split. The probabilities that are outside of the lattice are accumulated.

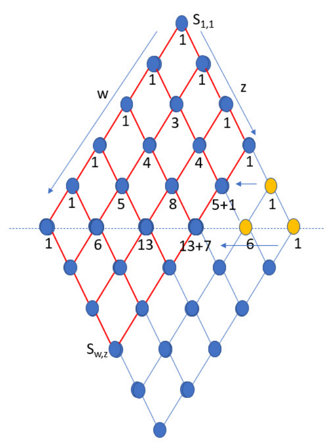

Fig. 5: Geographical Shortest Path Lattice Mapping on Pascal Triangle.

□

V. Simulation

This section details the simulation of messages sent by multiple terminals to a destination GW over a satellite grid, while handling handovers and seam disconnections. Multiple configurations of terminals on a grid of 8x8 satellites running for 10,000 cycles were executed. The SAGRW algorithm is compared against the location-based Routing algorithm. Table I summarizes the parameters and expected performance that will be simulated and evaluated for each algorithm. The virtual topology algorithm is added to the table for complexity reference.

The simulation demonstrates how a total demand of C (the link capacity) can be routed to the serving GW. The simulation is coded in Python and was executed on a dual core i5 Intel PC running Windows10 OS. The simulated scenarios include:

Property	**Virtual topology circuits**	**Location-based routing**	**SAGRW**
Terminal to GW routing over a moving satellite grid complexity and validation	$O(\log(number of flows))$	$O(1)$	$O(1)$
Routing complexity per handover event	$O(V+E)$	not supported	$O(1)$
Routing over seam	$O(V+E)$	not supported	$O(1)$

TABLE I: Simulated Algorithms

- Handover - The terminal(s) and GW handover from a serving satellite to a rising satellite due to vertical and horizontal movement of the constellation over the satellite grid.
- Seam disconnections - The seam barrier is simulated by failing all relevant east-west and west-east ISLs between the two sub-grids.

The simulation execution is divided into cycles. At the end of the execution, the demand (number of messages) received by the GW node should be the number of cycles multiplied by the demand sent on each cycle or number of cycles x ISL capacity. The following is executed on each cycle:

- Grid shift and handovers - The serving satellite of each terminal is calculated according to its location.
- Seam disconnection - the location of the seam is calculated and the relevant ISLs get disconnected.
- Traffic generation.
  - The traffic demand is split into messages (an ISL can transfer up to its capacity of messages in each cycle).
  - Each terminal is assigned with a demand.
- Message routing.
  - Each satellite enqueues the traffic (messages) from the terminals and from the ISLs (links).
  - Each satellite runs the algorithm the enqueued traffic.

The algorithms executed include location-based routing and SAGRW. The simulation executes the algorithms to route the traffic to the satellite closest to the destination terminal location. When reaching the closest satellite, the message is received by the GW/Terminal.

A. Latency and Capacity Analysis

The impact of handovers on the latency until the message is received by the GW is tested in the simulation by adding a 'round counter' to the message. The counter is evaluated when the message is received on the GW. The terminals are assigned to different locations. The coordinates range from 1.00 to 16.00 with a granularity of $1/50$. The min path and max path latency difference indicates the change in latency (by number of hops). In addition, latency changes and total capacity received on the GW by number of cycles is tested. The simulation results indicate that the latency was increased or decreased by one hop at the most (due to handover) and all traffic demand was received by the GW in case of SAGRW. In case of location-based routing, messages were received only when the two parties were located on the same sub-grid (no need to cross the seam).

B. Seam Disconnection Analysis

Figure 6 details the simulation of a location-based algorithm running on a polar constellation with a seam. Fifty simulation rounds of 2000 cycles were executed. In each simulation, two terminals were randomly located. One of the terminals is sending messages to the other at a rate of C. The figure details the disconnection time related to the seam (percentage), and compares it to the calculated seam cycle. The difference between the calculated and simulated graph is related to the handover impact (causing a temporary change of the relative terminals distance).

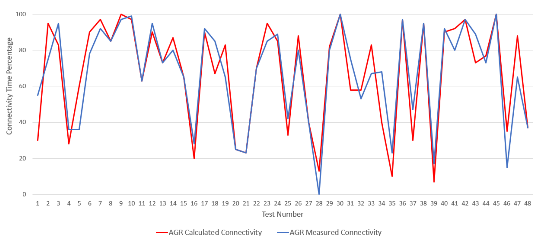

Fig. 6: Seam Cycle Analysis.

C. SAGRW Path Length

Figure 7 details the simulation of the SAGRW algorithm running on a polar constellation with a seam. Fifty simulation rounds of 2000 cycles were executed. In each simulation, two terminals were randomly located. One of the terminals is sending messages to the other at a rate of C. All the messages exchanged were received. The figure details the calculated path length versus the measured one. The difference between the calculated and simulated graph is related to the handover impact (causing a temporary change of the relative terminals distance).

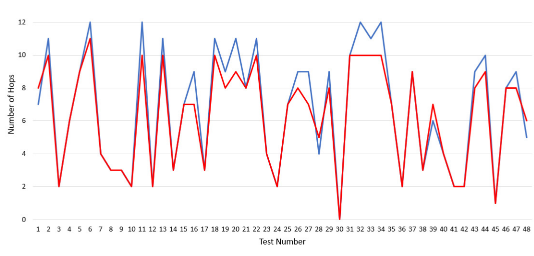

Fig. 7: SAGRW Path Length Analysis

VI. Discussion

SAGRW and AGR represent a unique approach to perform LEO constellation routing. These protocols provide very low complexity algorithms to route traffic from a terminal to an opposite terminal (or a GW) and do not require strict timing mechanisms (implicitly required in current work on virtual node and virtual topology). Both protocols provide access routing, support handover, reduce the last hop routing complexity and avoid the privacy concerns, all of which are not handled by current work [2], [11]. SAGRW builds a geographical shortest path lattice and uses random walk to distribute the load on the lattice. It handles handovers, seam disconnections, and supports any-to-any routing while not revealing the terminal location.

References

[1] S. H. Chang, W. B. Kim and G. C. Lee, "FSA-based link assignment and routing in low-earth orbit satellite networks," *IEEE transactions on vehicular technology*, pp. 47(3): 1037-1048, 1998.

[2] T. R. Henderson and R. H. Kat, "On Distributed, Geographi-Based Packet Routing for LEO Satellite Networks," *Proceedings of IEEE Globecom 2000*, Nov 2000. doi:10.1109/GLOCOM.2000.891311

[3] O. Korcak and F. Alagoz, "Analysis of priority-based adaptive routing in satellite networks," *The 2nd International symposium on wireless communication systems*, pp. 629-633, 2005.

[4] O. Korcak and F. Alagoz, "Priority-based Adaptive Shortest Path Routing in IP over LEO Satellite Networks," *23rd AIAA International communications satelllite systems Conference*, 2005.

[5] Krishnamoorthy, T. Andreas, Ernst and Mohan, "Solution algorithms for the capacitated single allocation hub location problem," *Annals of operations research 86*, p. 141–159, 1999.

[6] X. Liu, Z. Jiang and C. Liu, "A low-complexity probabilistic routing algorithm for polar orbits satellite constellation networks," *IEEE/CIC International conference on communications in China (ICCC)*, pp. 1-5, 2015.

[7] X. Liu, X. Yan and Z. Jiang, "A low-complexity routing algorithm based on load balancing for LEO satellite networks," *IEEE 82nd Vehicular technology conference*, pp. 1-5,2015.

[8] N. Shacam, "Protocol for Multi-Satellite Networks," *IEEE MILCOM*, pp. 501-505, 1988.

[9] H. Tan and L. Zhu, "A novel routing algorithm based on virtual topology snapshot in LEO satellite networks," *IEEE 17th International conference on computational science and engineering (CSE)*, pp. 357-361, 2014.

[10] S. A. N. Firouzja, M. Yousefnezhad, M. F. Othman and M. Samadi, "A wised routing protocols for LEO satellite networks," *Asian Control Conference (ASCC)*, pp. 1-6, doi: 10.1109/ASCC.2015.7244415, 2015.

[11] X Qi, J Ma, D Wu, L Liu, S Hu, "A survey of routing techniques for satellite networks," *Journal of Communications and Information Networks* , pp 66–85, 2016. doi: 10.1109/TMC.2018.2831679

[12] Wu, Z., Hu, G., Jin, F., Song, Y., Fu, Y., and Ni, G., "A novel routing design in the IP-based GEO/LEO hybrid satellite networks," *Int. J. Satell. Commun. Network*, 35: 179– 199. doi: 10.1002/sat.1174, 2017.

[13] Na Zhenyu, Pa Zheng, Liu Xin, Deng Zhian, Gao Zihe, Guo Qing, "Distributed Routing Strategy Based on Machine Learning for LEO Satellite Network," *Wireless Communications and Mobile Computing* 1-10., 10.1155/2018/3026405, 2018.

[14] J. Gómez-Gardeñes; V. Latora, "Entropy rate of diffusion processes on complex networks," *Physical Review E* 78 (6), 065102, 2010.

[15] Beraldi, Roberto, "Biased Random Walks in Uniform Wireless Networks," *IEEE Transactions on Mobile Computing* . 8 (4): 500–513, 2008.

[16] Oren Markovitz, Michael Segal, "Advanced Routing Algorithms for Low Orbit Satellite Constellations," *IEEE ICC*,2021.

[17] M. De Sanctis, E. Cianca and M. Ruggieri, "Ip-based routing algorithms for leo satellite networks in near-polar orbits," *IEEE Aerospace Conf.*, 2003, pp. 1273-1280, doi: 10.1109/AERO.2003.1235242.

[18] J. Wenjuan and Z. Peng, "An Improved Connection-Oriented Routing in LEO Satellite Networks," *WASE Int. Conf.*, 2010, pp. 296-299, doi: 10.1109/ICIE.2010.77.