A source routing algorithm based on CGR for DTN-Nanosatellite Networks

Mario Marchese and Fabio Patrone
Department of Electrical, Electronic, Telecommunications Engineering, and Naval Architecture (DITEN)
University of Genoa - Via All'Opera Pia, 13 - 16145 Genoa (Italy)
Email: mario.marchese@unige.it, f.patrone@edu.unige.it

***Abstract*—The number of nanosatellites orbiting around the Earth is increasing year after year. Nanosatellite constellations can be deployed to cover even larger areas. However, data exchange among nanosatellites is not trivial, especially due to the required hardware components related to the limited size and weight. Moreover, in some cases, contacts between nanosatellites and ground stations cannot always be guaranteed. The Delay and Disruption Tolerant Networking (DTN) paradigm allows storing data in nanosatellite and ground station buffers until the contact with the next hop is available. Routing in this kind of network is a crucial aspect. Delivery times are larger compared to a "classical" network due to the time that data have to wait inside intermediate node buffers and to the limitation of available resources, especially on-board nanosatellites. The adoption of a smart routing strategy can contribute relieving this gap. In this paper, we propose S-CGR, a Source routing algorithm based on the Contact Graph Routing (CGR). It computes a routing path from source to destination nodes for each bundle, which is the data unit in DTN networks. S-CGR considers static and known a priori information about contacts (begin times, end times, and overall contact volumes) and dynamic information about nanosatellite buffer occupancies and available contact volumes. The complete source/destination paths are stored in the bundles. Intermediate nodes read the routing instructions from the bundles without any computational effort.**

I. Introduction

In the last decade, a lot of satellites have been launched and are still orbiting around the Earth. Thanks to the Micro-Electronics (MEs) and Micro-Systems Technologies (MSTs), very small hardware components have been designed [1], thus reducing satellite size, weight, and cost, and leading to an increase in the number of entities, such as small industries, universities, government of small countries, interested to gain access to space. These small satellites are called nanosatellites. The changing budgetary and operational environment has triggered interest in nanosatellites in a variety of research fields and sectors. Military and intelligence services are looking at smaller-sized satellites to create a highly-responsive, launch-on-demand constellation. On the science side, for example, several efforts highlight the utility of small spacecrafts for in-situ atmospheric research. A widely employed kind of nanosatellite is called CubeSat [2]. The great attraction of this product is the possibility to create an entire satellite by choosing all the hardware components that better fulfil the target mission and keep low construction and launch costs.

Hundreds of projects which involve nanosatellites are on-going [3] and have different aims, such as weather monitoring, disaster prevention, and space observation in fields such as astronomy, atmospheric science, biology, Earth observation, and telecommunications [4]. These projects involve the deployment of a single or a group of nanosatellites structured in swarms or constellations. Constellations are composed of tens or hundreds of nanosatellites which can be spread in different orbital planes. Communications can take place between two nanosatellites in the same orbital plane (intra-orbit) or in different orbital planes (inter-orbit). A peculiar feature of nanosatellite communication is that the links among nanosatellites, as well as the links between nanosatellites and ground stations, could not be always active. The lack of a persistent path between source and destination must be tackled in the design of a nanosatellite infrastructure. The DTN paradigm [5] deals with link disruptions and long delays, allowing nodes (both ground stations and nanosatellites) to store data until the next contact is available [6]. However, even if this feature deals with the problem of not persistent paths, the time needed to complete data delivery could be very high, especially compared to the one of a "classical" network such as the Internet. To perform a smart routing strategy helps reduce this time. Moreover, a suitable routing strategy should also consider the reduced available resources, especially on-board nanosatellites, such as storage capacity, energy, and computational power.

We propose a source routing algorithm based on Contact Graph Routing (CGR), called S-CGR, aimed at reducing resource consumption in terms of computational load and obtained delivery time.

The paper is structured as follows: the next section lists the routing challenges over DTN-Nanosatellite networks; Section III presents S-CGR; Section IV contains the performance evaluation; and Section V the conclusions.

II. Routing Challenges in DTN-Nanosatellite networks

In a nanosatellite network, a topical aspect is that there may be no persistent paths between sources and destinations because satellite links are not always up. The routing problem in DTN may at first appear as the standard problem of dynamic routing extended to link failures, but it is not so. For the standard dynamic routing problem, the topology is assumed to be connected and the objective of the routing algorithm is to find the best currently-available path to move traffic end-to-end. In DTN an end-to-end path may be permanently unavailable, so routing is performed over time to achieve data delivery by employing long-term storage at intermediate nodes. The DTN routing problem is a constrained optimization problem where single links may be unavailable for a long time and with storage constraints at each node. The problem is simplified by the fact that satellites move in a predictable way, so information about contact start times and durations is known a priori. Moreover, given the bandwidth, and, consequently, the data rate of satellite links, also the amount of data that can be exchanged during each contact is known. One of the most used routing algorithms in DTN networks where changes in connectivity are planned and scheduled is the Contact Graph Routing (CGR) algorithm [7]. Routing decisions are performed by using a "Contact Plan", which is a time-ordered list of scheduled contacts indicating start and end time of each contact. Each node between the source and the destination calculates its own path for each packet, called bundle, and locally makes its own next hop decision. There are papers in literature whose purpose is to prove the reliability of CGR in Low Earth Orbit (LEO) satellite communication networks [8] [9]. Other papers propose to extend the basic version of CGR through enhancements which can be useful for routing in the considered DTN-Nanosatellite network. For example, a version called CGR-ETO (Earliest Transmission Opportunity) has been proposed in [10]. It considers also the queuing delay due to bundles already stored and waiting to be transmitted. An additional problem needs to be addressed in case of different priority classes. When a node forwards a higher priority bundle it deliberately neglects lower priority ones previously received in order to enforce priority. This can cause oversubscription of a contact if the contact is already fully booked by lower priority bundles. A modification of CGR, called Overbooking Management, is proposed in [11]. It minimizes the consequences of overbooking by early handling. Another extension consists in a source routing version of the CGR called CGR with Extension Blocks (CGR-EB) [12]. It encodes path and traffic information into routed messages, extending CGR in two ways: (1) it permits non-monotonically increasing or decreasing cost functions and (2) it uses virtual circuits to avoid routing loops and reduce computations [13]. Similar studies related to routing in this kind of networks have been performed by the authors in [14]–[16]. In these studies, we have focused our attention on the bundle delivery time reduction as the only considered performance parameter. With this work, we also consider another performance parameter, the computational load due to the routing process and its reduction obtainable by using a source routing algorithm.

III. Algorithm Description

The algorithm proposed in this paper, called S-CGR, is a source routing version of the CGR algorithm inspired to the CGR-EB proposed in [13]. Source DTN nodes compute a routing path for each bundle through the entire DTN portion by using CGR. Intermediate DTN nodes only forward the bundle following the already calculated path. This reduces the resource consumption in terms of computational load because intermediate nodes avoid next hop computation. A source DTN node computes a routing path for a bundle destined to a destination DTN node by using information about future contacts. This information is included in the Contact Plan which is locally stored in each DTN node and concerns, for S-CGR, start and end times, and available contact volumes (amounts of data that can still be exchanged). After computing the routing path, the source DTN node updates its local Contact Plan in order to ignore a saturated contact in next routing path computations.

A Routing Extension Block has been defined to store the routing path inside the Bundle Header, following the framework defined in [17]. Its structure is shown in Figure 1.

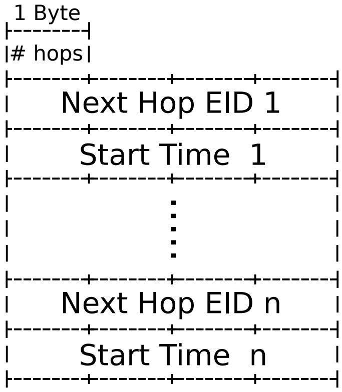

Figure 1: Routing Extension Block

The first field (*# hops*, 8 *bit*) indicates the number of routing path's hops. After that, a list of n Node Endpoint Identifiers (EIDs) (*Next Hop EIDs*, 32 *bit*) and n time values (*Start Times*, 32 *bit*) uniquely identify the sequence of contacts that the bundle has to exploit. If an intermediate node is not able to forward the bundle following the routing instructions, it misses one of the planned transmission opportunities and acts as described in the following. The intermediate nodes periodically perform a validation process reading the start time sequence in order to realize if the information in the bundles' Routing Extension Block is still useful. If it is not, routing paths are re-computed and Bundle Headers updated. This situation may be due to: a) Transmission losses, quite frequent especially in satellite links. b) Errors in the routing path computation. c) Each node computes a routing path for each bundle it has to send by using the information in its local Contact Plan. If the paths computed for different bundles by different nodes have some links in common, the amount of allocated contact volumes of these links could exceed their overall values, and the nodes will not be able to forward all scheduled bundles before the contacts end. DTN nodes have to forward information about Contact Plans through the entire DTN network in order to avoid the computation of erroneous paths. d) Another parameter which could lead to routing path recalculation is the nanosatellite buffer occupancy. A nanosatellite can obviously receive and store a bundle only if it has enough free space in its buffer. This is another information a nanosatellite has to send to the node in contact before starting the bundle transmission in order to avoid losses due to buffer overflow.

To summarize, the operative steps of S-CGR are:

1) DTN source nodes compute a routing path for each bundle from DTN source to DTN destination by using CGR, store this information in the defined Bundle Header extension block, called Routing Extension Block, and send the bundles.
2) At the beginning of each contact, ground stations and nanosatellites send a Status Bundle to the node in contact containing information about Contact Plan changes (if any). This information will be used in the routing path validation process and in case of

routing path recalculation. Nanosatellites also send a second Status Bundle containing information about their buffer occupancies. This information will be used in the verification process.

3) DTN intermediate nodes that receive a bundle read the proper next hop EID in the Routing Extension Block and forward the bundle as soon as possible after verifying if the next hop has enough free space in its buffer (verification process).

4) After the end of all contacts, the DTN intermediate nodes verify, for each bundle stored in their buffers, if the contact with the next hop concerns a future contact or a past one (validation process). The second case means that the bundle cannot be forwarded through the planned contact and so a new routing path is needed.

IV. Performance Analysis

To test the proposed algorithm, we consider a possible application scenario focusing on the telecommunication purpose, and, in particular, on the employment of a nanosatellite constellation to extend network access to rural and remote areas, as proposed in [18]. Ground stations located in rural areas, called Cold Spots (CSs), collect data from rural terminals, called Rural nodes (Rs), to deliver to the Internet through the nanosatellite constellation. Ground stations located in urban areas and directly linked to the Internet, called Hot Spots (HSs), exchange data on one side with nanosatellites and on the other side with Internet servers.

Figure 2 shows a schematic 2-D representation of the mentioned DTN-Nanosatellite network scenario: rural users or devices (R_1, …, R_N, R_{N+1}, …, R_M) are linked with the proper Cold Spot (CS_1 or CS_2). Nanosatellites (SAT_1, SAT_2, and SAT_3) upload and download data from ground stations, both Cold Spots and Hot Spots (HS_1 and HS_2) changing their position along a defined low orbit. C is the central control station, called Central Node, which manages the all network and D is an Internet Server. Links between Rural nodes and Cold Spots, between Hot Spots and Central Node, and between Central Node and Internet Servers are permanently active: DTN is not required. Links between Cold Spots and nanosatellites and between nanosatellites and Hot Spots are not permanently active: communications from/to nanosatellites rely on the DTN. The considered nanosatellite constellation is composed of more than one orbital plane, so data can also be exchanged among nanosatellites through inter-satellite links.

We have developed a module for the software Network Simulator 3 (NS3). It includes:

- **a Scenario module**: it allows setting different network parameters in order to simulate different scenarios;
- **a DTN module**: it implements the characteristics of the DTN paradigm needed to perform a communication in this DTN-Nanosatellite network. It includes store and forward mechanism, a personalized and light version of the Bundle Protocol [19], and the S-CGR algorithm;
- **a LEO nanosatellite constellation module**: it computes and updates the position of each nanosatellite during the simulation time.

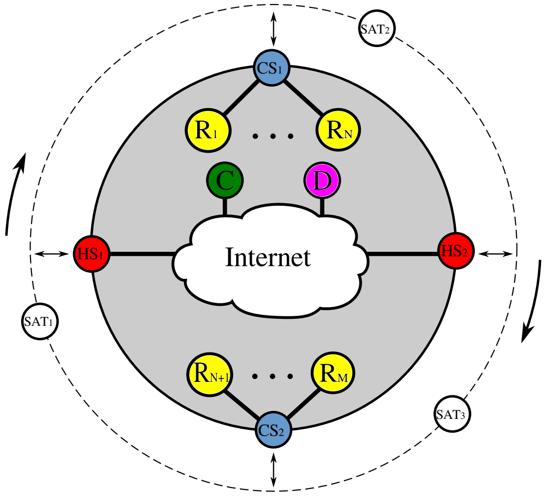

Figure 2: Nanosatellite network scenario.

The simulator allows setting the number of ground stations and nanosatellites and their positions in a 3-D space. Ground station positions are established through their Latitude, Longitude, and Altitude (LLA) coordinates. Given the number of nanosatellites, the number of orbits, and the desired orbital parameters, nanosatellite positions are computed and updated by using the widespread orbital model called NORAD SGP4 [20]. In our tests, all Nanosatellites are equally distributed among circular orbits and equally spaced within each orbit. Inter-satellite links are allowed only among nanosatellites of different and adjacent orbital planes (only inter-orbit satellite links).

We performed a set of simulations fixing the number and the geographic position of each rural area (in particular of each Cold Spot) in order to evaluate the obtained performances changing the number of Hot Spots ($nHSs$), nanosatellites ($nSATs$), and orbital planes ($nORBITs$). 10 rural areas are used. The number of Internet Servers and Rural Users per rural area are also fixed to 100 and 1 000, respectively.

Table I shows the 16 simulated scenarios. Figure 3 highlights the ground station positions on the Earth's map, evidencing the used $nHSs$ and, in brackets, the geographic location referenced in Figure 3, $nSATs$, and $nORBITs$.

Simulated traffic flows are composed of service requests generated by Rural nodes and service replies generated by Internet Servers. 10 000 requests are generated from each Rural Area for each scenario with a uniform generation function during all simulation duration. The Rural node which generates the request is chosen randomly with uniform distribution, as well as the destination Internet Server. Since Rural nodes and Internet servers are not DTN nodes, they generate request and reply packets which are forwarded to Cold Spots and Central Node, respectively, by using the standard TCP/IP protocol stack. Cold Spots and Central Node encapsulate packets in bundles which are routed through the DTN portion of the network. The numerical values used for nanosatellite constellation, satellite communication, and traffic flow configuration are summarized in Table II.

The performance in terms of resource consumption and delivery time obtained by using the S-CGR algorithm has been compared with the one offered by the standard CGR.

Table I: Simulated scenarios

		$nSATs - nORBITs$			
		10 − 1	**20 − 2**	**40 − 4**	**80 − 8**
$nHSs$	**6 (1st set)**	Scenario 1	Scenario 2	Scenario 3	Scenario 4
	12 (1st + 2nd sets)	Scenario 5	Scenario 6	Scenario 7	Scenario 8
	18 (1st + 2nd + 3rd sets)	Scenario 9	Scenario 10	Scenario 11	Scenario 12
	24 (1st + 2nd + 3rd + 4th sets)	Scenario 13	Scenario 14	Scenario 15	Scenario 16

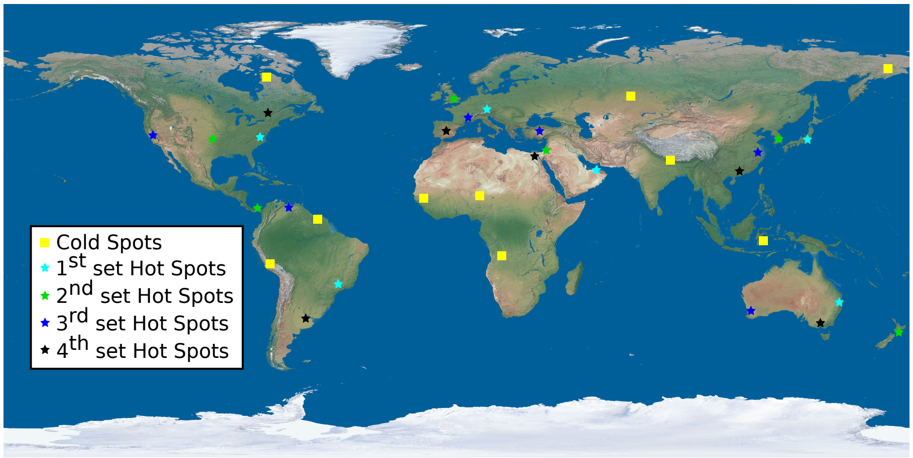

Figure 3: Ground Station positions on the Earth's map

Table II: Nanosatellite constellation, satellite communication, and traffic flow configuration parameters

Eccentricity	0
Semi-major axis	600 km
Inclination	88°
Right ascension of the ascending node	$0^\circ, \ldots, \frac{o-1}{O} * 360^\circ \quad o = 1, \ldots, O$ O: number of orbits
Argument of perigee	90°
True Anomaly	$0^\circ, \ldots, \frac{n-1}{N} * 360^\circ \quad n = 1, \ldots, N$ N: number of nanosatellites per orbit
Minimum Elevation angle for GS ÷ SAT transmission	20°
Frequency carrier	2228 MHz (S-band)
Uplink and Downlink transmission rate	1 $Mbps$
Request Size	8 kB
Reply Size	1 MB
Nanosatellite storage capacity	1 GB
Simulation Duration	24 hr

One of the performance metrics is the average number of times the routing algorithm has been executed to compute the next hop per bundle. The obtained results show that the proposed source routing algorithm allows saving resources, especially in terms of computational load and, consequently, energy consumption. This improvement is shown in Figure 4.

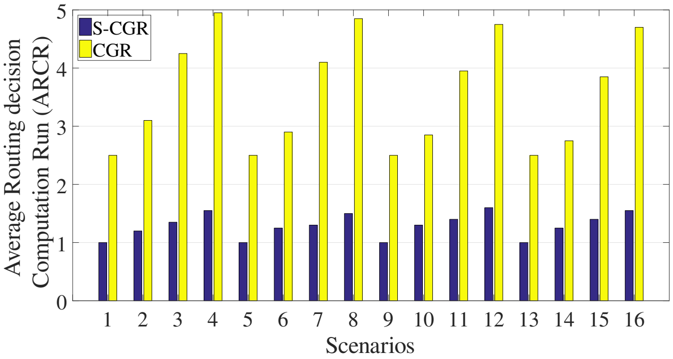

Figure 4: Comparison between Average Routing decision Computation Runs (ARCRs) obtained by using S-CGR and CGR

The ARCR reduction obtained by using S-CGR is ranging between 54% and 69%. Moreover, the number of routing decision computation runs performed by nanosatellites by using S-CGR are much less than the one obtained by using the standard CGR. They are ranging between 0% and 19% by using S-CGR and between 40% and 68% by using the standard CGR. Resource saving mainly takes place in the nodes where available resource limitations are stronger, i.e. the nanosatellites.

The second performance metric is the Average Delivery Time (ADT) defined as:

$$ADT = \frac{\sum_{r=1}^{R} \left(T_r^{RX} - T_r^{TX}\right)}{R} \qquad (1)$$

R is the total number of requests generated by the Rural Nodes during the simulation, T_r^{TX} and T_r^{RX} are the time instants when the r^{th} request is sent and the related reply is received by the Source Rural node, respectively.

The obtained results are shown in Figure 5.

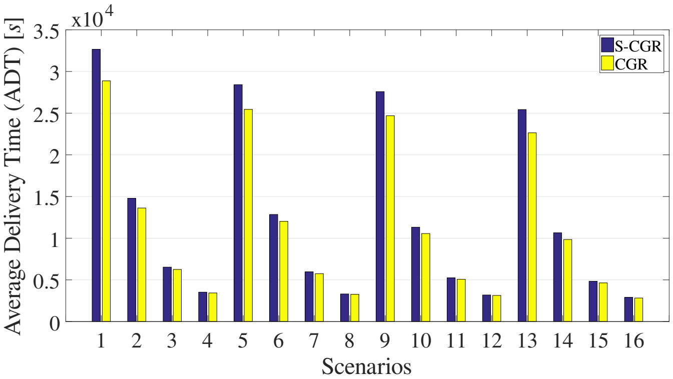

Figure 5: Comparison between Average Delivery Times (ADTs) obtained by using S-CGR and CGR

The ADT increases by using S-CGR from 2% to 13%. The reason is that the overall size of the bundles increases by adding the Bundle Extension Block, essential for S-CGR implementation. For example, considering the worst case (Scenario 4) where ARCR is almost 5 for CGR and about 1.5 for S-CGR, the Bundle Extension Block size is 41 Byte. This value is negligible compared to the reply size (0.0041%), but it is not compared to the request size (4.1%). As a consequence, the number of request bundles sent per contact decreases leading to an increase of the ADT. Increasing the number of nanosatellites, the ADT values tend to be the same (see, for example, Scenarios 4, 8, 12, and 16 in Figure 5), because also the number of contacts between ground stations and nanosatellites, and, consequently, the ground station upload capacities, increase.

V. Conclusions and future works

In this paper we have presented S-CGR, a CGR-based source routing algorithm able to reduce resource consumption in terms of computational load related to the routing process in DTN-Nanosatellite networks. S-CGR exploits static and known a priori information related to the contacts between nanosatellites and ground stations and among nanosatellites, and dynamic information related to nanosatellite buffer occupancies. A possible application scenario which involves a nanosatellite constellation to extend network access to rural and remote areas has been considered in the performance evaluation phase. Simulations have been performed changing the topology of the network (the number of nanosatellites and ground stations) and modelling traffic flows as requests and replies of a generic application. Rural nodes send request packets to Internet servers which forward back reply packets to source Rural nodes.

Results have been collected in terms of the average number of times the routing algorithm has been executed to compute the next hop per bundle (Average Routing decision Computation Runs ARCRs) and Average Delivery Time (ADT) both by using S-CGR and the standard CGR. S-CGR assures a decrease of the ARCRs and, consequently, of the computational load "paid" by an increase of the average delivery time due to the larger size of the S-CGR Bundle Header, which includes the source to destination path together with contact start times.

To perform other simulations changing parameters such as request size, reply size, and satellite link transmission rate in order to identify a possible relation between Bundle Header overhead and delivery time could be an interesting future work. It would be interesting to find out the maximum Bundle Header overhead that allows using S-CGR without any delivery time increase. Moreover, a larger set of dynamic information could be considered in the routing path computation, including, for example, the available energy, which is strictly limited onboard nanosatellites.

References

[1] E. Gill, G. Monna, J. Scherpen, and C. Verhoeven, "Misat: Designing a series of powerful small satellites based upon micro systems technology," in *58th International Astronautical Congress*, vol. 6, 2007, pp. 1–6.

[2] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "Cubesat: A new generation of picosatellite for education and industry low-cost space experimentation," 2000.

[3] Nanosatellite database. [Online]. Available: http://www.nanosats.eu/

[4] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Advances in Space Research*, vol. 47, no. 4, pp. 663–684, 2011.

[5] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," *RFC4838*, 2007.

[6] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, "Delay-and disruption-tolerant networking (dtn): an alternative solution for future satellite networking applications," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1980–1997, 2011.

[7] S. Burleigh, "Contact graph routing," *Internet Draft*, 2010.

[8] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, "Contact graph routing in dtn space networks: overview, enhancements and performance," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, 2015.

[9] C. Caini and R. Firrincieli, "Application of contact graph routing to leo satellite dtn communications," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3301–3305.

[10] N. Bezirgiannidis, F. Tsapeli, S. Diamantopoulos, and V. Tsaoussidis, "Towards flexibility and accuracy in space dtn communications," in *Proceedings of the 8th ACM MobiCom workshop on Challenged networks*. ACM, 2013, pp. 43–48.

[11] N. Bezirgiannidis, C. Caini, D. P. Montenero, M. Ruggieri, and V. Tsaoussidis, "Contact graph routing enhancements for delay tolerant space communications," in *Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014 7th*. IEEE, 2014, pp. 17–23.

[12] E. J. Birrane, "Improving graph-based overlay routing in delay tolerant networks," in *Wireless Days (WD), 2011 IFIP*. IEEE, 2011, pp. 1–6.

[13] E. Birrane, S. Burleigh, and N. Kasch, "Analysis of the contact graph routing algorithm: Bounding interplanetary paths," *Acta Astronautica*, vol. 75, pp. 108–119, 2012.

[14] M. Cello, M. Marchese, and F. Patrone, "Hotsel: A hot spot selection algorithm for internet access in rural areas through nanosatellite networks," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015.

[15] ——, "Satsel: A satellite selection algorithm to reduce delivery time in dtn-nanosatellite networks for internet access in rural areas," in *8th Advanced Satellite Multimedia Systems Conference and the 14th Signal Processing for Space Communications Worksho (ASMS/SPSC), 2016*. IEEE, 2016.

[16] ——, "Coldsel: A selection algorithm to mitigate congestion situations over nanosatellite networks," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016.

[17] E. Birrane, "Contact graph routing extension block," *Internet Draft*, 2013.

[18] S. Burleigh, "Nanosatellites for universal network access," in *Proceedings of the 2013 ACM MobiCom workshop on Lowest cost denominator networking for universal access*. ACM, 2013, pp. 33–34.

[19] K. L. Scott and S. Burleigh, "Bundle protocol specification," *RFC5050*, 2007.

[20] D. Vallado, P. Crawford, R. Hujsak, and T. Kelso, "Revisiting spacetrack report# 3," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006, p. 6753.