DTN-Based Nanosatellite Architecture and Hot Spot Selection Algorithm for Remote Areas Connection

Mario Marchese ⓘ , *Senior Member, IEEE*, Fabio Patrone, *Member, IEEE*, and Marco Cello, *Member, IEEE*

Abstract—To extend the network coverage to rural and remote areas, there are different solutions with their advantages and disadvantages. We propose to use a nanosatellite network and to exploit the ability to cope with large delays and disruptions provided by the delay- and disruption-tolerant networking (DTN) paradigm. The connection from remote areas is managed through ground stations called cold spots that collect data from rural nodes and address them to the nanosatellites. Nanosatellites carry and download data to hot spots that address them to the Internet destinations. On the reverse direction, Internet data are addressed to hot spots, uploaded on nanosatellites and delivered to the rural destinations through cold spots. The problem of choosing the "optimal" hot spot to address data from the Internet source is important because a wrong choice could lead to large delivery delays. In this paper, we propose "gRANteD": a Nanosatellite-DTN Network for rural and remote areas. gRANteD includes "HotSel," a hot spot selection algorithm to minimize the delivery time of all data destined to rural users. The performance evaluation is carried out through a DTN module which implements gRANteD and HotSel, within the framework of Network Simulator 3.

Index Terms—DTN, nanosatellite communications, remote areas connection, satellite networks, TCP/IP.

I. INTRODUCTION

A. The Use of Nanosatellites for Network Access in Rural Areas

THE digital divide in the world is still evident [1]. One of the reasons is that many countries do not have a suitable ICT infrastructure to provide network connection at least in given periods of the day for specific non-real-time services. Satellites and Multi-Layered Satellite Networks [2] can be an alternative solution but current technologies require high costs in the construction, launch, and maintenance. Nanosatellites [3] have been recently proposed as a cost-effective solution to extend

the network access in rural and remote areas. The main reasons for the development and use of nanosatellites are, for now: enabling low data rate communications, gathering data from multiple points, and inspecting the activities of larger satellites. CubeSat [4], which requires 0.1% of the cost of a classical LEO satellite, is aimed at enabling a constellation of nanosatellites for Earth imaging even if other applications are not excluded. Nanosatellites main reference applications are non-real time services. Low data rate real-time service (e.g., web browsing) might be provided at cost of a large number of nanosatellites in the constellation and of Earth stations, as also shown in this paper. Otherwise, it is possible to refer to a sort of delay-tolerant web browsing where a larger delay than the one of regular web services may be tolerated.

B. Introduction to gRANteD and HotSel

In this scenario we would like to get a transparent bidirectional communication between the users in the rural areas and the servers on the Internet. Communications must be based on the TCP/IP protocol stack running on endpoints in order to avoid software modifications on Internet servers and to make rural endpoints real hosts of the Internet. To reach this aim we propose "gRANteD", a novel architecture able to guarantee a transparent communication among TCP/IP endpoints through a Delay and Disruption Tolerant Networking (DTN)-based heterogeneous network. Differently from Geostationary Earth Orbit (GEO) satellite networks, in nanosatellite networks there is hardly or never a persistent communication path between rural nodes and destination servers, because all links may be not up at the same time [5]. The DTN architecture [6] provides long term information storage at intermediate nodes so tackling link disruptions, very long delays, and intermittent connectivity. The action is carried out through an overlay protocol, called Bundle Protocol (BP) [7], developed on top of either transport (such as TCP and UDP) and lower layer (such as Bluetooth and Ethernet) protocols. BP data unit is called "bundle", which is a message that encapsulates application layer protocol data units.

Accordingly, gRANteD performance is heavily affected by the time that bundles must wait in ground stations and nanosatellites buffers. On the path from rural nodes to the Internet (forward path), data delivered to the cold spot managing the remote region and uploaded to nanosatellites are downloaded to the first available hot spot along the path. Even if the dynamic choice of

uploading nanosatellites and downloading hot spots might bring advantages to the overall communication, the forward path is not considered for optimization in this paper. On the reverse path, a server on the Internet that wants to reply to a rural user has some hot spot alternatives to whom it can deliver data. Different hot spots can send data to the destination with different delivery delays depending on the number, position, and buffer occupancy of the nanosatellites they come in contact with. The problem of choosing the "best" hot spot impacts on the delivery delay. To tackle this issue we introduce "HotSel", a hot spot selection mechanism, part of the gRANteD architecture, able to minimize the delivery time.

C. Structure of the Paper

The remainder of this paper is structured as follows. Section II describes the state of the art. Section III presents the reference scenario of a nanosatellite-DTN-based network to connect remote areas. Section IV describes the gRANteD architecture. The proposed HotSel algorithm is described in Section V. Sections VI and VII provide, respectively, a description of the Bundle Protocol extension necessary for gRANteD and HotSel and of the DTN solution implemented in the NS3 simulator. The performance analysis is shown in Section VIII. Conclusions are drawn in Section IX.

II. STATE OF THE ART

Connecting remote areas where there is no terrestrial infrastructure or where it is damaged to the Internet is not a recent challenge. [8] proposes a method to establish a communication with the Internet for the nomadic Saami population by using DTN mobile devices. [9] introduces DakNet, an ad-hoc wireless network that provides asynchronous connectivity based on rural kiosks and portable storage devices mounted on a bus, a motorcycle or a bicycle. A similar architecture, again based on mechanic backhauls, is described in [10]. The architecture used in [11] is a multi-hop mesh network composed of long-distance 802.11 links. A communication system based on the combination of UAVs, DTN wireless technologies, and regular traffic of boats has been proposed in [12]. [13] exploits public transportation and commuting vehicles such as taxis equipped with wireless DTN-enabled devices. Some of the described architectures offer valid and inexpensive solutions (e.g., with an investment of $15 million, the mentioned DakNet [9] could equip 50 000 rural buses in India), but may suffer from performance limits and insecurities due to the use of mechanic backhauls.

Concerning emergency communication systems in case of large-scale disasters that can damage and temporarily turn off the telecommunication infrastructure: [14] proposes a multi-hop Device-to-Device (D2D) communication network based on DTN/MANET terminals; [15] exploits the use of balloons, electric vehicles, and UAVs; and [16] proposes to use a Disaster Information Network System based on DTN with Cognitive Wireless Network (CWN). Enlarging Internet access is very useful also for healthcare in rural areas [17], [18].

The Internet Research Task Force (IRTF) created a working group called Global Access to the Internet for All (GAIA) in 2014 [19] aimed at increasing the interest on the challenges and opportunities in enabling global Internet access. Projects related to GAIA, including Project Loon and Internet.org mentioned below, are summarized in [20]. Project Loon [21] is an initiative by Google which involves the use of balloons in the stratosphere, at an altitude of about 20 km. Internet.org [22] is a project started by Facebook in partnership with other companies, non-profit organizations, and governments. Internet.org goal is to create a wireless network by using Free Space Optics and radio waves.

Other projects of main interest are: architectuRE for an Internet For Everybody (RIFE) [23]; Aerial Base Stations with Opportunistic Links for Unexpected and Temporary Events (ABSOLUTE) [24]; SpaceX and partners [25]; and OneWeb [26].

Existing commercial satellite networks may be an alternative to connect remote areas to the Internet. Iridium [27], Globalstar [28], and Orbcomm [29] are LEO satellite constellations that provide phone and data communication services, such as Iridium Certus, the new service platform to be powered by the new Iridium NEXT constellation [30]. Inmarsat [31] is a GEO satellite constellation for voice and data services. These solutions offer a complete coverage, they do not involve disruptive links and may be used to offer also real-time services. Even if performances are good, they are very expensive due to the production and launch costs of satellites. For example, after its complete deployment, the Iridium NEXT network will be composed of 81 satellites able to offer a transmission rate of up to 1.4 Mbps in L-band frequency with an estimated cost of about $3 billion. The Inmarsat's Global Xpress (GX) constellation is formed of four Ka-band, high-speed mobile broadband communications GEO satellites, with a cost of $1.6 billion.

III. NANOSATELLITE-DTN-BASED NETWORK FOR RURAL ACCESS: REFERENCE SCENARIO

The joint use of nanosatellites and DTN paradigm [3] allows avoiding the drawbacks of mechanic backhauls and assures lower costs than GEO and LEO existing commercial satellite networks even if, as said in the introduction, it can't currently provide a continuous overall coverage and, consequently, real-time services.

Fig. 1 shows the reference scenario (partially taken from [3]) of a Nanosatellite-DTN network.

The following types of nodes are used and depicted in Fig. 1.
1) *Rural nodes* (R_1, ..., R_N, R_{N+1}, ..., R_M): represent the users located in rural and remote areas.
2) *Cold Spots* (CS_1 and CS_2): are the ground stations that, on one hand, collect all data from rural nodes in order to upload them on nanosatellites, and, on the other hand, download data destined to rural users from nanosatellites.
3) *Hot Spots* (HS_1 and HS_2): are the ground stations that, on the forward direction, download data from nanosatellites and forward them to the destination nodes in the Internet, and, on the reverse direction, upload data on the nanosatellites that will deliver them to the destination rural nodes.
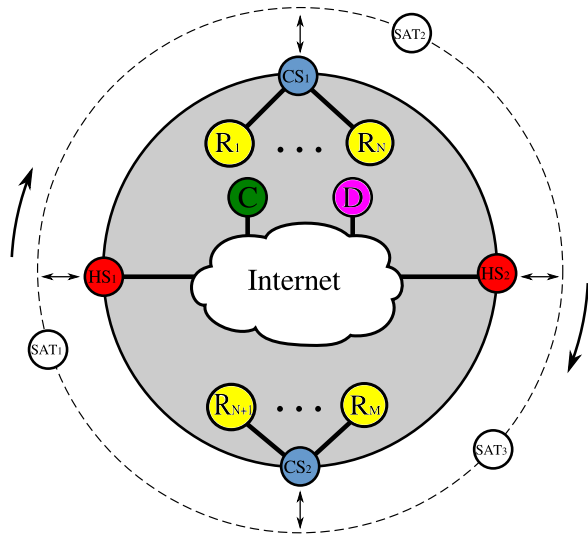
Fig. 1. Nanosatellite network scenario.

4) *Nanosatellites (SAT_1, SAT_2, and SAT_3):* are the nanosatellites which exchange data with ground stations (both hot and cold spots). They change their position along their orbit.

5) *Internet node (D):* is an Internet node (e.g., Database, Mail, Web Server, Remote Controller, ...) which acts as the destination of all rural nodes requests.

6) *Central node (C):* its purpose is to receive all rural nodes requests in order to collect all responses and to forward each of them back to the proper hot spot. In practice C is the decision maker that will implement the HotSel algorithm within the gRANteD architecture.

When a rural user, for example R_1, wants to connect to the Internet, it sends its request to the cold spot (e.g., CS_1) that manages the remote region where it is located. CS_1 waits the transit of the next nanosatellite (SAT_1 in the example in Fig. 1). If the nanosatellite has sufficient storage, it accepts and transports the request. When it comes in contact with the first hot spot on its route (HS_2), it delivers the request to it. No dynamic choice of the hot spot where to deliver forward requests is forecast in this paper: the request is sent to the first hot spot met along the path. HS_2 forwards the message to the central node C. C reads the message to know which is the destination node (e.g., D) and delivers the request to it. On the reverse path, D sends back the response to the central node. C selects one of the hot spots under its control and forwards the response to it. The selected hot spot uploads the message on a nanosatellite that transfers it to the proper cold spot which finally delivers the response to the rural node R_1.

IV. GRANTeD ARCHITECTURE

Links between cold spots and nanosatellites and between nanosatellites and hot spots are not permanently active satellite links. Relying the communications from/to nanosatellites on DTN tackles this situation. The gRANteD architecture and related nodes are shown in Fig. 2.

Cold spots have to connect with both rural hosts and nanosatellites, so these nodes must have both a TCP/IP and a DTN interface. Hot spots are completely DTN nodes because their purpose is only to forward the bundles received from nanosatellites to the central node and to upload all the bundles sent by the central node on nanosatellites. The central node has both a TCP/IP interface, which allows it to establish a TCP connection with the destination node in order to collect all the information required by the source node, and a DTN interface aimed of creating and forwarding bundles to hot spots. The connection is split into three parts, as clear in Fig. 2. The highest layer of the Central Node and of all Cold Spots is indicated as *Application**. It is a particular relay layer extracting information from bundles and, if needed, uniquely associating each received request to its source node. The central node has the same characteristics. It is also important to remark the relay action of the Bundle Layer [5] at hot spot and cold spot nodes, which allows not only the information storage at intermediate nodes but also the implementation of dedicated protocol stacks for single network portions, as happens over the satellite links in Fig. 2. The satellite-dedicated protocol stack has been left unspecified in Fig. 2. Actually it can include, if necessary, a satellite-specific transport, network, and data link layer, as done in this paper (Section VII) for implementation reasons, but, given the nature of point-to-point connectivity in this case and the chance to use the routing functions included in the Bundle Layer, the use of a satellite data link layer may be enough.

V. HOT SPOT SELECTION

A. HotSel Implementation

In this section we describe "HotSel", the dynamic hot spot selection method implemented in the central node C.

HotSel computes the optimal hot spot choice to minimize the delivery time of each bundle destined to rural users. To do this, the central node needs to know, in each time instant, the current position of nanosatellites and hot/cold spots (this information is supposed to be known in this paper[1]), and how hot spots and nanosatellites buffer occupancy will evolve, in order to predict how much data will be loaded on each satellite and when the new bundle will be served. In particular, at each bundle arrival, for each "candidate" hot spot, the algorithm computes the number of nanosatellites necessary to upload the already queued bundles and the new arrived one. To fulfil this aim, HotSel analyses each nanosatellite in order to estimate its buffer occupancy evolution in the near future. No other parameter is considered by HotSel to make the hot spot choice as simplest as possible. The literature contains papers, partially from the same authors, about the choice of the Earth Station where to access the satellite from a sensor network [33], where other metrics possibly contrasting with each other such as packet loss rate, average packet delay, and energy consumption are considered by using the Multi-Attribute Decision Making (MADM) environment.

---

[1]as also done by the Contact Graph Routing (CGR) [32] which may be considered the state of the art concerning LEO satellite networks and where decisions are made by using only information about contacts, which depend on the number and position of satellites and hot/cold spots
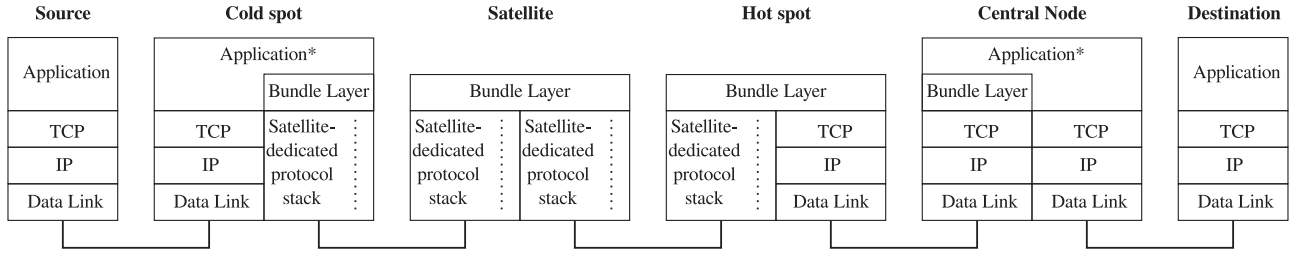
Fig. 2.   Network architecture.

Input: B_J - bundle destined to CS_J; $\mathcal{HS}, \mathcal{CS}, \mathcal{SAT}$ - set of hot spots, cold spots and nanosatellites, respectively;
Output: i^o - HS_i that minimizes the delivery time of bundle B_J;

1 **foreach** hot spot $i \in \mathcal{HS}$ **do**
2     $d_{i,J}^{\text{HS}} \leftarrow d_{i,J}^{\text{HS}} + 1$;
3     $S_i \leftarrow ComputeNumberNanosatellites\ (d_{i,J}^{\text{HS}})$;
4     $D_{i,J} \leftarrow w_{i,k} + (S_i - 1)\dfrac{W}{N_{SAT}} + t_{i,J}$;
5 $i^o = \underset{i \in \mathcal{HS}}{\operatorname{argmin}}\, D_{i,J}$;

Fig. 3.   HotSel algorithm.

The use of other metrics, which requires to take measures or to estimate metric values through theoretical models, needs an attentive analysis with respect to time, computational load, and implementation issues also considering current Bundle Protocol [7] standard, and may be considered for future research.

A preliminary description and evaluation of HotSel can be found in [34] and [35]. Similar studies related to routing have been performed in [36] and [37].

HotSel algorithm is shown in Fig. 3. A bundle B_J needs to be transmitted to CS_J, $J \in \mathcal{CS}$, being $\mathcal{CS} = \{1, N_{CS}\}$ the set of cold spots identifiers and J the index identifying a specific element within CS. For each hot spot HS_i, $i \in \mathcal{HS}$, being $\mathcal{HS} = \{1, N_{HS}\}$ the set of hot spots identifiers and i the index identifying a specific element within HS, HotSel computes the delivery time $D_{i,J}$ needed to transmit B_J using HS_i as hot spot. The number of bundles $d_{i,J}^{\text{HS}}$ queued on HS_i and directed to CS_J is updated considering the new arrived bundle B_J. The new value is $d_{i,J}^{\text{HS}} \leftarrow d_{i,J}^{\text{HS}} + 1$. Then the function $ComputeNumberNanosatellites$ computes the number S_i of nanosatellites on the constellation that HS_i uses to upload the queued data ($d_{i,J}^{\text{HS}}$) destined to CS_J.

The delivery time $D_{i,J}$ is computed as:

$$D_{i,J} = w_{i,k} + (S_i - 1)\frac{W}{N_{SAT}} + t_{i,J}, \qquad (1)$$

where $w_{i,k}$ is the flight time from its current position to HS_i of the first nanosatellite SAT_k that will come in contact with i, $k \in \mathcal{SAT}$, being $\mathcal{SAT} = \{1, N_{SAT}\}$ the set of nanosatellite identifiers and k a specific nanosatellite index. N_{SAT} is the overall number nanosatellites. Being W the overall revolution time of each nanosatellite, $\frac{W}{N_{SAT}}$ is the average flight time between two nanosatellites. $\frac{W}{N_{SAT}}$ is the flight time of each SAT_k, $\forall k \in \mathcal{SAT}$, between its contact with HS_i and CS_J. HotSel iterates on all hot spots. The optimal $i^o - th\ HS\ (HS_{i^o})$ that

---

1 **function** $ComputeNumberNanosatellites\ (d_{i,J})$;
2 nanosatellite $k \leftarrow$ first nanosatellite comes in contact with HS_i;
3 $S_i = 0$ **repeat**
4     $nextHS(k)$: next HS on the path of SAT_k;
5     $\mathcal{CS}_k$: subset of $\mathcal{CS}$ on the path of SAT_k between $nextHS(k) - 1$ and $nextHS(k)$;
6     $UploadDataOnSAT\ (k, nextHS(k), \mathcal{CS}_k)$;
7     $S_i \leftarrow S_i + 1$;
8     $k \leftarrow k - 1$;
9 **until** $d_{i,J}^{HS} = 0$;
10 **return** S_i;

Fig. 4.   Function ComputeNumberNanosatellites.

---

1 **function** $UploadDataOnSAT\ (k, nextHS(k), \mathcal{CS}_k)$;
2 **if** $nextHS(k) = i$ **then**
3   **if** $\mathcal{CS}_k \neq \emptyset$ **then**
4        $d_{k,j}^{\text{SAT}} = 0\ \forall j \in \mathcal{CS}_k$;
5     $calculate\ p_{i,j,k}\ using\ Eqs.\ (3) - (5)$;
6     $d_{i,j}^{\text{HS}} \leftarrow d_{i,j}^{\text{HS}} - p_{i,j,k},\ \forall j \in \mathcal{CS}$;
7     $d_{k,j}^{\text{SAT}} \leftarrow d_{k,j}^{\text{SAT}} + p_{i,j,k},\ \forall j \in \mathcal{CS}$;
8 **else**
9   **if** $\mathcal{CS}_k \neq \emptyset$ **then**
10        $d_{k,j}^{\text{SAT}} = 0\ \forall j \in \mathcal{CS}_k$;
11     $calculate\ p_{l,j,k}\ using\ Eqs.\ (8) - (10)$;
12     $d_{l,j}^{\text{HS}} \leftarrow d_{l,j}^{\text{HS}} - p_{l,j,k},\ \forall j \in \mathcal{CS}$;
13     $d_{k,j}^{\text{SAT}} \leftarrow d_{k,j}^{\text{SAT}} + p_{l,j,k},\ \forall j \in \mathcal{CS}$;
14
15     $nextHS(k) \leftarrow nextHS(k) + 1$;
16     $\mathcal{CS}_k \leftarrow$ subset of $\mathcal{CS}$ on the path of SAT_k between $nextHS(k) - 1$ and $nextHS(k)$;
17     $UploadDataOnSAT\ (k, nextHS(k), \mathcal{CS}_k)$;

Fig. 5.   Function UploadDataOnSAT.

minimizes the delivery time of the bundle B_J is:

$$i^o = \underset{i \in \mathcal{HS}}{\arg\min}\, D_{i,J}. \qquad (2)$$

B. Function ComputeNumberSatellites

The implementation of the function $ComputeNumber$ $Nanosatellites$ is reported in Fig. 4.

SAT_k is selected as the first nanosatellite that will come in contact with HS_i (Line 2). The function enters in a do-while loop in which SAT_k is virtually moved on its path, also downloading data to cold spots and uploading data from hot spots until it comes in contact with HS_i. The action of uploading from cold spots and downloading to hot spots, belonging to the forward channel actions, is neglected in this paper, as said before. At the contact the amount of HS_i queued data to be uploaded will depend on the residual transport capacity of

SAT_k. SAT_k movement and relative data upload from hot spots and download to cold spots is simulated by using the function $UploadDataOnSAT$ shown in Fig. 5 and explained in the next section. If SAT_k comes in contact with other hot spots and cold spots during the virtual movement, the relative queues are updated according to the data stored in the nodes. Lines 4 and 5 in Fig. 4 define two variables: $nextHS(k)$, which represents the next hot spot with which SAT_k will come in contact, and $\mathcal{CS}_k$, which is the set of cold spots that are located between the two hot spots where SAT_k is currently positioned. We indicate with $nextHS(k) - 1$ the hot spot before $nextHS(k)$ on the clockwise orbit path. Referring to Fig. 1 and assuming SAT_k as SAT_3, it is true that $nextHS(k) = HS_1$, $\mathcal{CS}_k = CS_2$ and $nextHS(k) - 1 = HS_2$. When SAT_k has virtually received the data from HS_i and from the other hot spots, the variable S_i is incremented by 1 and the next nanosatellite is analysed ($k \leftarrow k - 1$ means that the nanosatellite behind k is analysed). The loop terminates when all the bundles queued in HS_i and destined to CS_J have been virtually uploaded ($d_{iJ} = 0$). Obviously, at the end of the loop, the variable S_i contains the number of nanosatellites necessary to complete this operation.

C. Function UploadDataOnSAT

The aim of the function $UploadDataOnSAT$ is to simulate the movement of SAT_k along its path and the consequent data upload/download, having HS_i, $nextNS(k)$, and CS_k as input.

In case the next hot spot with which SAT_k comes in contact is just the analysed HS_i (as HS_1 in Fig. 1 with SAT_1), the lines 3-7 in Fig. 5 are processed. If there is at least one cold spot in the orbit portion in which SAT_k is located ($\mathcal{CS}_k \neq \emptyset$), SAT_k downloads all the data destined to those cold spots and then sets the number of bundles stored on SAT_k and destined to CS_j to zero: $d_{k,j}^{\text{SAT}} = 0$, $\forall j \in \mathcal{CS}_k$. The assumption that a generic nanosatellite can download all data destined to a given cold spot simplifies the computations and the description of the algorithm but introduces a constraint on the amount of bundles that can be uploaded on the satellite itself, given the "download to cold spots/upload from hot spots" transmission rates and the time window at each nanosatellite transit. We suppose this amount of data fixed and equal to Q. This means that, in each orbit round, each SAT_k cannot carry more than Q bundles from hot spots and can download to the destination cold spot only this amount of data. In practice, from the operative viewpoint, the behaviour is the same as if each nanosatellite had separated onboard buffers for destination cold spot and each buffer had a maximum length of Q bundles.

After this operation, SAT_k is virtually moved until it comes in contact with HS_i. HotSel computes the number $p_{i,j,k}$ of bundles destined to CS_j that HS_i will upload on SAT_k as follows:

$$p_{i,j,k}, \qquad \forall j \in \mathcal{CS} \tag{3}$$

s.t.

$$p_{i,j,k} \leq \min[d_{i,j}^{HS}, Q - d_{k,j}^{SAT}], \tag{4}$$

$$\sum_{\forall j \in \mathcal{CS}} p_{i,j,k} \leq Q; \tag{5}$$

Constraints 4 and 5 mean, respectively, that: 1) HS_i cannot upload more data than those it has stored and it cannot upload more than $Q - d_{k,j}^{SAT}$ data destined to CS_j. In this way SAT_k will not carry more than Q data destined to CS_j and it will empty the buffer dedicated to CS_j when it will come in contact with CS_j. 2) The total amount of data that HS_i uploads to SAT_k is bounded by Q.

Finally, hot spot and nanosatellite buffer occupancies are updated:

$$d_{i,j}^{HS} \leftarrow d_{i,j}^{HS} - p_{i,j,k}, \qquad \forall j \in \mathcal{CS} \tag{6}$$

$$d_{k,j}^{SAT} \leftarrow d_{k,j}^{SAT} + p_{i,j,k}, \qquad \forall j \in \mathcal{CS}. \tag{7}$$

Alternatively, if the next hot spot with which SAT_k comes in contact is not HS_i (Lines 8-17 in Fig. 5), as HS_1 in Fig. 1 with SAT_2, HotSel proceeds making the calculations described before but considering the next hot spot with which SAT_k comes in contact, called HS_l. In particular:

$$p_{l,j,k}, \ \forall j \in \mathcal{CS} \tag{8}$$

s.t.

$$p_{l,j,k} \leq \min[d_{l,j}^{HS}, Q - d_{k,j}^{HS}], \tag{9}$$

$$\sum_{\forall j \in \mathcal{CS}} p_{l,j,k} \leq Q; \tag{10}$$

The equations above are derived from (3), (4), and (5) with the substitution of i with l. Then HS_l and SAT_k buffer occupancies are updated:

$$d_{l,j}^{HS} \leftarrow d_{l,j}^{HS} - p_{l,j,k}, \qquad \forall j \in \mathcal{CS} \tag{11}$$

$$d_{k,j}^{SAT} \leftarrow d_{k,j}^{SAT} + p_{l,j,k}, \qquad \forall j \in \mathcal{CS}. \tag{12}$$

Finally, SAT_k is virtually moved to the next orbit portion (it would be moved after HS_2 in Fig. 1): $nextHS(k)$ and $\mathcal{CS}_k$ are updated and $UploadDataOnSAT$ is called recursively.

To make this algorithm usable in any nanosatellite network topology is necessary that:

1) the overall buffer of each hot spot has always enough free space to store the messages received from the central node;
2) the overall buffer of all nanosatellites is big enough in order to allow storing the maximum possible amount of data destined to each cold spot; it practice, it has to be at least $Q \cdot N_{HS}$ bundles.

VI. NEEDED EXTENSION TO THE BUNDLE PROTOCOL

Given the requirements of HotSel, it is necessary to define a proper extension of the standard version of the Bundle Protocol [7] in order to handle the following aspects:

1) Nanosatellites must forward status information about the number of bundles stored on board for any destination cold spot;
2) Hot spots must periodically forward status information about the number of stored bundles for any destination cold spot to the central node.

In practice, it is necessary to introduce a special status bundle which contains hot spot and nanosatellite buffer occupancy information (i.e., the status information) used to correctly implement the hot spot selection. The status bundle may be practically implemented in the extension blocks envisioned in [7]. In this view, the implementation of HotSel is fully compliant with the original BP standard.

VII. DTN IMPLEMENTATION IN NS3

The aim is to test gRANteD and HotSel over a simulated nanosatellite network scenario. The developed simulator is based on the software Network Simulator 3 (NS3). We have implemented an additional module which includes a personalized and lighter version of the Bundle Protocol, an implementation of the main characteristics of the DTN paradigm, functions related to the LEO satellite movement, the proposed gRANteD architecture, and the HotSel algorithm. Due to the operative limitations of the used simulation tool concerning nanosatellite and DTN environments, some approximations have been carried on to pursue the aim.

A. gRANteD Architecture Implementation

Explicitly referring to Fig. 2: links between Destinations and Central Node are not modelled. The two nodes are the same in the simulations. The Central Node is the destination of all requests and the origin of the return channel.

The links between Cold Spots and Rural Nodes and between Hot Spots and Central Node use the following protocols: TCP, IP, and Ethernet. They are simulated in NS3 by using NS3 TCP, IP, and CSMA modules. The sender node opens a transmission TCP socket for each transmitting bundle. It is so possible to achieve multiple contemporary bundle transmissions.

The satellite-dedicated protocol stack acting over the links belonging to the Space Segment is approximated as follows. An overall block composed of transport, network, and data link protocols has been chosen to simplify the implementation over NS3. Concerning the data link layer, we adapt the NS3 Wi-Fi module opportunely setting some parameters in order to allow the transmission at distances much bigger than the ones of standard Wi-Fi networks. We made this choice because even though several Satellite Data Link layer protocols have been defined [38], no one is available in NS3. Since the focus of this work is not related to the design and test of a satellite data link protocol, we can use the NS3 Wi-Fi module in this simulation environment without affecting the obtained performance trend. We have defined a couple of Wi-Fi interfaces (one for the receiver and another one for the transmitter) for both Nanosatellites and Ground Stations. In this way, these nodes are able to exchange bundles even though they are located hundreds of kilometres from each others, avoiding problems related to the carrier sense. The Satellite Network layer protocol is IP, which is fully implemented through the NS3 IP module. For the Satellite Transport layer there are different possible protocols [39], [40]. We have chosen the UDP, available in the NS3 UDP module. UDP is simple, connectionless, and introduces a few bytes of overhead due to its reduced header size and a limited complexity and

resource utilization. The drawback is that UDP offers unreliable communications: if a bundle were corrupted or lost during a satellite communication, there would be no way to identify this loss and to retransmit the bundle. To avoid this, we have implemented a DTN acknowledgement mechanism within the modified Bundle Layer described in the following.

The "change" of transport protocol on all ground stations is really implemented through the relay action of the Bundle Layer. Hot spots remove TCP headers from the data received by the central node and encapsulate the received bundles in UDP to upload them on nanosatellites. In rural areas, cold spots do the reverse operation removing UDP headers and encapsulating bundles over TCP to forward them to rural nodes. Above the transport layer, we have implemented a DTN module for NS3, starting from [41].

B. Bundle Protocol Implementation

The implemented DTN layer assures the main functionalities of the bundle protocol but bypasses the following actions: Late Binding, Multicast, Priority Classes, and Security.

The modified bundle protocol header is composed as follows:
1) Destination Endpoint ID (EID), 32 *bit*: text string which identifies the destination DTN node;
2) Source Endpoint ID (EID), 32 *bit*: text string which identifies the source DTN node;
3) Sequence Number, 32 *bit*: number which differentiates each bundle created by a DTN node from all the other ones created by the same DTN node;
4) Payload Size, 32 *bit*: size of the DTN payload in *Byte*;
5) Source Timestamp, 32 *bit*: time instant when the bundle has been created;
6) Type, 8 *bit*: number which distinguishes the bundle type:
   a) 0 – data bundle: it contains information data;
   b) 1 – ack bundle: it is used to have a reliable transmission over satellite links and to implement the custody transfer function;
   c) 2 – status bundle: it contains the status information as specified in Section VII-A used to correctly implement the hot spot selection.

The number of bundles destined to specific cold spots and stored within each nanosatellite k ($d_{k,j}^{SAT}, \forall j \in \mathcal{CS}$ and $\forall k \in \mathcal{SAT}$) and hot spot i ($d_{i,j}^{HS}, \forall j \in \mathcal{CS}$ and $\forall i \in \mathcal{HS}$) are addressed to the Central Node in the payload of status bundles (Type=2) sent at the beginning and at the end of each contact between hot spots and nanosatellites.

Concerning the DTN acknowledgement mechanism operating over UDP-based satellite portions, when a node sends a data bundle through the satellite link, the receiver sends back a DTN ack bundle to notify the successful reception. The sender keeps a copy of sent bundles until it receives the DTN ack for all these bundles. This action not only adds reliability to satellite links but, in practice, is the way we use to implement the custody of a bundle so bypassing the real implementation of the custody transfer mechanism.

In more detail, refering to the bundle protocol header, if a data bundle (Type=0) with a given Sequence Number arrives at

a DTN node, the bundle is acknowledged by using an ack bundle (Type=1). This ack bundle transports the acknowledgement number within the Sequence Number field. Status bundles are not acknowledged. To practically implement the mechanism, each node keeps a list of its active transmission sockets. At a new transmission, the sender node creates and adds a new socket to the list (a socket for each bundle for design choice), sets a timeout, and, after the end of the transmission (or after the reception of the DTN ack bundle), removes the socket from this list. If the socket is still in the list when the timeout expires, it means that the communication is not successfully completed and the data bundle is sent again to the output buffer.

To allow each intermediate DTN node to forward bundles, we have implemented a round-robin scheduling policy. All DTN nodes store each bundle in a different queue depending on the destination and send the first bundle of each queue, serving the queues in circular order. In this way, it will be possible in the future to assign a different priority to each queue, in order to implement a sort of QoS mechanism. From the operative viewpoint, we manually set the routing tables of all DTN nodes. To simulate traffic flows we don't use traffic applications defined in NS3, but directly a set of bundles that are enqueued in the output buffer.

VIII. PERFORMANCE ANALYSIS

We define as orbit portion each part of the orbit between two consecutive hot spots. Nanosatellites are considered equally spaced assuming that the orbit is almost circular and the nanosatellites speed is constant even though in a real scenario it is not exactly so. The simulation environment represents 2-D scenarios. The first set of tests, in Section VIII-A, focuses on the performance of HotSel when compared with two alternative algorithms by varying the bundle generation function and the network load configuration. The number of hot spots, cold spots, and nanosatellites is kept low to allow better focusing on the decision mechanisms used by HotSel with respect to the other schemes. The performance of HotSel are checked by increasing the satellite transmission rate in Section VIII-B and the number of nanosatellites and hot spots in Section VIII-C. Increasing the number of these elements allows reducing the end-to-end delay to values that may be compatible with regular web browsing. Checking the feasibility of web browsing in this environment is the aim of Section VIII-D. Section VIII-E analyses the nanosatellite buffer occupancy obtained through HotSel. The trade-off between network performance and costs is the object of Section VIII-F.

A. Definition of Performance Metrics and HotSel Comparison With Static and Random Choices

We performed a set of tests by using two different scenarios:
1) *Scenario 1:* composed of 4 hot spots (HS_1-HS_4), 4 nanosatellites (SAT_1-SAT_4), 8 cold spots (CS_1-CS_8), and 2 rural nodes for each cold spot (R_1 and R_2 are linked to CS_1, R_3 and R_4 are linked to CS_2, ...). Its topology is shown in Fig. 6.
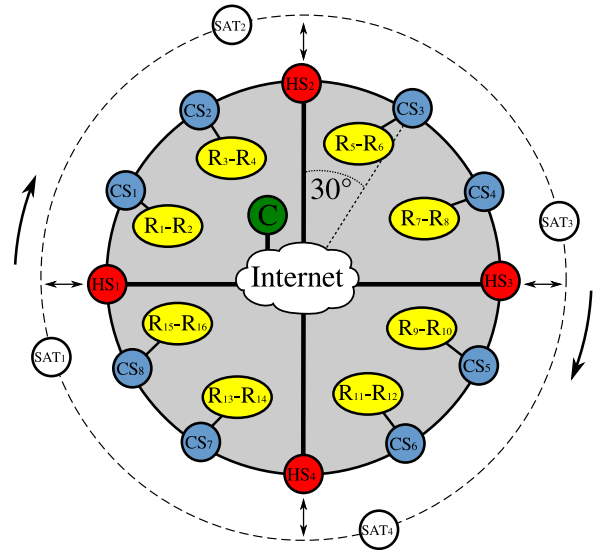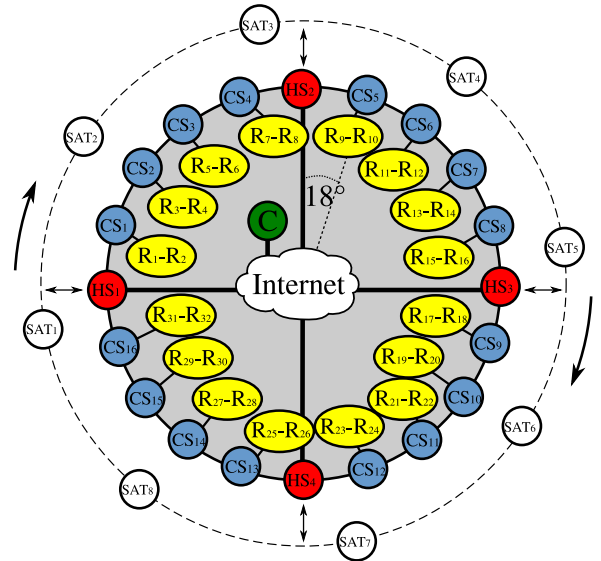

Fig. 6. Scenario 1.

Fig. 7. Scenario 2.

2) *Scenario 2:* composed of 4 hot spots (HS_1-HS_4), 8 nanosatellites (SAT_1-SAT_8), 16 cold spots (CS_1-CS_{16}), and 2 rural nodes for each cold spot (R_1-R_{32}). Its topology is shown in Fig. 7.

For each scenario, we have performed different simulations by varying the bundles generation function of each traffic flow and the group of destination rural nodes, in order to test the performance in different realistic traffic load situations and to individuate a group of exemplary load cases that may impact on performance. Each simulated traffic flow has the central node as source node and a rural node as destination node. For the sake of implementation simplicity, the application layer is not implemented and the bundle layer acts directly as traffic generator. It obviously implies that the rural destination nodes implement the bundle layer instead of the application layer. The following formalism is used to define the rural nodes where the traffic is

TABLE I
SATELLITE COMMUNICATION AND TRAFFIC FLOW
CONFIGURATION PARAMETERS

orbit shape	circular
satellite altitude	200 km
satellite velocity	7 788 m/s
satellite coverage radius	500 km
orbit time (W)	5 300 s
transmission range	500 km
uplink and downlink transmission rate	230 kbps
contact duration	256 s
contact volume (Q)	7 MB
bundle size	100 kB
flow size	500 bundles
bundle generation interval	$1 \div 500$ s

TABLE II
ADT [S] WITH UNIFORM BUNDLE GENERATION FUNCTION

		1P	2CP	2NCP	AP
Scenario 1	Static Choice	4 648	4 648	4 648	1 961
	Random Choice	3 693	4 315	4 316	4 009
	HotSel	3 421	3 800	3 413	1 961
Scenario 2	Static Choice	2 388	2 388	2 388	894
	Random Choice	2 898	3 182	3 187	2 893
	HotSel	1 986	2 195	1 986	894

addressed: the notation R_x-R_y-R_z-R_t means that the bundles are addressed to R_x, R_y, R_z, and R_t at the same time.

The numerical settings regarding the satellite communication and the traffic flow configurations are summarized in Table I.

We have used four different bundle generation functions. n is the bundle index that varies from 1 to 500. T_n is the time instant in seconds when the n-th bundle is generated. T_n ranges in the same time interval for the different generation functions to fairly compare the performance:

1) *Uniform*:

$$T_n = n \qquad (13)$$

2) *Fast start*:

$$T_n = \left(\frac{500 + 9 * n}{510 - n} \right) \qquad (14)$$

At the beginning of the simulation each flow has a high transmission rate which decreases rapidly in the first half of the simulation and slowly in the second half.

3) *Slow start*:

$$T_n = \left(\frac{510 * n - 500}{n + 9} \right) \qquad (15)$$

At the beginning of the simulation each flow has a low transmission rate which increases slowly in the first half of the simulation and rapidly in the second half.

4) *On/Off*:

$$T_n = \left(\left\lfloor \frac{n-1}{100} \right\rfloor * 100 \right) + \frac{\left((n-1) \bmod 100 \right)}{2} + 1 \qquad (16)$$

Each traffic flow generates a bundle every 0.5 s in time intervals (in s) [1, 50.5], [101, 150.5], [201, 250.5], [301, 350.5], [401, 450.5] and does not transmit any bundles in the other time intervals, mimicking the behaviour of data bursts.

The parameter used to evaluate the performance is the Average Delivery Time (ADT) defined as:

$$ADT = \frac{\sum_{m=1}^{M} ADT_m}{M} \qquad (17)$$

where

$$ADT_m = \frac{\sum_{n=1}^{N_m} \left(T_n^{RX} - T_n^{TX} \right)}{N_m} \qquad (18)$$

M is the number of traffic flows in the simulation, N_m is the number of bundles of the m-th traffic flow, T_n^{RX} is the time instant when the n-th bundle is received, and T_n^{TX} is the time instant when the n-th bundle is transmitted by the source.

For each simulation, we have compared the ADT from the central node to the rural destination nodes for three different mechanisms of hot spot selection:

1) *HotSel*.
2) *Static choice*: all bundles destined to a specific rural area are forwarded to the last hot spot met by nanosatellites before the reference cold spot of the rural area (for example, all bundles destined to R_1-R_2 will be forwarded to HS_1 for both scenarios). This action depends only on the contacts, i.e., on the number and position of nanosatellites and ground stations. It is the same choice made by the Contact Graph Routing (CGR) algorithm [32], widely used in satellite networks and in research studies as a comparison [42].
3) *Random choice*: the Hot Spot choice is random for each bundle with the same a priori probability for all rural areas.

To better quantify the performance improvement achievable by using HotSel, we have decided to test four different network load configurations for each scenario:

1) *One portion (1P)*: all traffic flow destination nodes are located in the same orbit portion. For Scenarios 1 and 2, 1P means to use the configurations named R_1-R_3 and R_1-R_5, respectively.
2) *Two consecutive portions (2CP)*: all traffic flow destination nodes are located in two consecutive orbit portions. For Scenarios 1 and 2, 2CP means to use the configurations named R_1-R_3-R_5-R_7 and R_1-R_5-R_9-R_{15}, respectively.
3) *Two not consecutive portions (2NCP)*: all traffic flow destination nodes are located in two not consecutive orbit portions (in our scenarios they are opposite portions). For Scenarios 1 and 2, 2NCP means to use the configurations named R_1-R_3-R_9-R_{11} and R_1-R_5-R_{17}-R_{21}, respectively.
4) *All portions (AP)*: the traffic flow destination nodes are equally distributed among all orbit portions. For Scenarios 1 and 2, AP means to use the configurations named R_1-R_5-R_9-R_{13} and R_1-R_9-R_{17}-R_{25}, respectively.

TABLE III
ADT [s] WITH FAST START BUNDLE GENERATION FUNCTION

		1P	2CP	2NCP	AP
	Static Choice	4 867	4 866	4 866	2 163
Scenario 1	**Random Choice**	3 306	4 513	4 469	4 120
	HotSel	3 093	3 674	3 092	2 163
	Static Choice	2 607	2 607	2 607	1 104
Scenario 2	**Random Choice**	2 813	3 437	3 411	3 045
	HotSel	1 999	2 289	2 004	1 104

TABLE IV
ADT [s] WITH SLOW START BUNDLE GENERATION FUNCTION

		1P	2CP	2NCP	AP
	Static Choice	5 669	5 669	5 669	3 008
Scenario 1	**Random Choice**	4 172	5 328	5 312	4 999
	HotSel	3 988	4 461	3 927	3 008
	Static Choice	2 789	2 789	2 789	1 307
Scenario 2	**Random Choice**	3 027	3 612	3 604	3 294
	HotSel	2 203	2 482	2 216	1 307

TABLE V
ADT [s] WITH ON/OFF BUNDLE GENERATION FUNCTION

		1P	2CP	2NCP	AP
	Static Choice	4 673	4 673	4 673	1 970
Scenario 1	**Random Choice**	3 543	4 299	4 289	3 981
	HotSel	3 434	3 823	3 436	1 970
	Static Choice	2 413	2 413	2 413	910
Scenario 2	**Random Choice**	2 836	3 182	3 187	2 893
	HotSel	2 024	2 220	2 024	910

Tables II–V show that, for all scenarios, all bundles generation functions, and all network load configurations, HotSel outperforms both Static and Random Choice selections.

The performance got by HotSel is very good for two reasons: 1) when the "nearest" hot spot to the destination cold spot is congestion free, HotSel chooses this hot spot; 2) if there is a congestion situation at the "nearest" hot spot to the destination cold spot, HotSel can upload on nanosatellites bundles belonging to all traffic flows through all hot spots, so increasing the amount of data carried by each nanosatellite during each orbit. This obviously cannot be done both by Static and by Random choice.

Tables II–V also highlight that in simulations $1P$, $2CP$, and $2NCP$ of Scenario 1, the worst performance is provided by Static Choice, while, for the same situations of Scenario 2, the worst result is got by Random Choice. The reason of this behaviour is that also Random Choice can take advantage of the possibility to upload on nanosatellites bundles through all hot spots, but the choice is not smart. On one hand, the amount of data destined to the same destination node that each nanosatellite can upload during each orbit is larger than the one available by using the Static Choice, on the other hand Random Choice does not choose the "nearest" hot spot to the destination cold spot

TABLE VI
PERCENTAGE PERFORMANCE INCREASE BY USING HOTSEL COMPARED WITH
STATIC AND RANDOM CHOICE

	1P	2CP	2NCP	AP
Static Choice	25%	15%	25%	0%
Random Choice	17%	23%	31%	56%

TABLE VII
ADT [s], DESTINATION RURAL NODES CHOSEN RANDOMLY

	Static Choice	Random Choice	HotSel
Scenario 1	12 000	8 940	7 600
Scenario 2	5 500	5 430	3 750

whenever possible, so nullifying the larger amount of available data.

In all cases, Random Choice does not take the optimal choice for all bundles and does not consider the presence of congestion situations, so it always offers worse performance than HotSel. Static Choice provides excellent performance when the destination nodes are equally distributed among the orbit portions (AP) but it is inefficient in all other distribution cases.

The mean overall performance increase offered by HotSel compared with Static and Random Choice for the different load configurations is shown in percentage in Table VI.

A real scenario can be seen as a sequence of time intervals of different length in which load configurations change unpredictably. To simulate this, we have performed 50 simulations where 10 traffic flows send bundles to rural nodes. Each flow consists of 500 bundles sent one per second. The tests have been repeated also for flows composed of 1 000, 1 500, and 2 000 bundles but the trend of the results does not change. For the sake of simplicity we show only the 500 bundles results. Each destination is chosen randomly, changing the seed of the random variable in each simulation. Also the starting time of each flow is chosen randomly in the time interval between 0 and 1 000 s. ADTs are shown in Table VII for the two used scenarios and the different selection schemes.

HotSel outperforms both Static and Random Choice. The percentage gain concerning the ADT provided by HotSel is: 37% compared with Static Choice, and 15% compared with Random Choice for Scenario 1; 32% compared with Static Choice, and 30% compared with Random Choice for Scenario 2. Even if these percentages are objectively meaningful and the improvement clear, it is important to practically evidence the advantage of using HotSel: in Scenario 1 the ADTs passes from 12 000 s (3 hours and 20 minutes) for the Static Choice and from 8 940 s (2 hours and 29 minutes) for the Random Choice to 7 600 s (about 2 hours and 6 minutes). Even if the order of magnitude of all these numerical values (that, moreover, do not include forward delays) are not compatible with regular Internet browsing, "saving" 1 hour and 24 minutes in a case and 23 minutes in the second case may be meaningful if a data exchange is on. I could get important news minutes before, or get

TABLE VIII
ADT [s] WITH DIFFERENT SATELLITE TRANSMISSION RATE

		1P	2CP	2NCP	AP
1 Mbps	**Scenario 1**	2 302	2 302	2 302	935
	Scenario 2	1 344	1 344	1 344	508
2 Mbps	**Scenario 1**	1 241	1 241	1 241	403
	Scenario 2	812	812	812	242
5 Mbps	**Scenario 1**	705	705	705	374
	Scenario 2	542	542	542	213
10 Mbps	**Scenario 1**	696	696	696	364
	Scenario 2	532	532	532	203

information, for example, to repair a given tool or monitor a parameter minutes before, so saving time and, probably, money. Similar comments may be reported for Scenario 2 where the delay passes from 5 500 s for the Static Choice and slightly less for the Random Choice to 3 750 s provided by HotSel. Again more than 1 hour to get data is not compatible with regular web browsing but the "half an hour gain" provided by HotSel may be very useful for many practical applications. To get lower delay values is necessary to increase the satellite transmission rate as well as the number of nanosatellites and hot spots, as shown in the remainder of the paper.

B. Increasing the Satellite Transmission Rate

Table VIII shows the ADT values for HotSel as in Table II (uniform distribution) but by setting uplink and downlink transmission rate to 1, 2, 5, and 10 Mbps.

Looking at the values got through a transmission rate of 5 and 10 Mbps in Scenario 2 suggests that the increase of the bandwidth together with the number of nanosatellites, hot spots, and cold spots could help get delay values almost compatible with Internet web browsing.

Of course, increasing the transmission rate and keeping fixed the load, the "nearest" hot spot to the destination cold spots is always congestion free and HotSel always chooses this hot spot. So the performance of HotSel and Static Choice are completely overlapped. The performance of the Random Choice ranges from about 3 200 s to 2 184 s (got for 10 Mbps, Scenario 2, AP).

C. Increasing the Number of Nanosatellites and Hot Spots

Fig. 8 shows the behaviour of the delivery time by varying the number of employed nanosatellites and hot spots but keeping constant the traffic load configuration and generation function (8 cold spots, 8 traffic flows of 1000 bundles each, 8 different destination rural areas, uniform traffic generation function).

Raising the nanosatellite and hot spot number, the performance improves. This is true because increasing the number of nanosatellites[2], the distance between two consecutive nanosatellites decreases, and increasing the number of hot spots, the amount of data that can be uploaded on nanosatellites during

each orbit grows up. For example, setting to 4 the nanosatellite number and changing the hot spot number from 1 to 5 the percentage performance improvement is about 80% (i.e., the percentage variation is −80%); setting to 1 the hot spot number and changing the nanosatellite number from 6 to 8, the percentage performance improvement is 21%, as highlighted through the first two ellipses of Fig. 8(a). The obtained ADT reduction fixing the hot spot number and varying the number of nanosatellites from 1 to 18 is about 92%, while fixing the nanosatellites number and varying the number of hot spots from 1 to 10 is about 88%. Fig. 8 also highlights that increasing the number of hot spots the percentage variation remains almost constant independently on the nanosatellite number: increasing the number of hot spots from 1 to 5 the percentage performance increase is always about 80% by using a different number of nanosatellites. The cases for 4, 10, and 18 nanosatellites are evidenced in Fig. 8(a) through vertical ellipses. Similarly, increasing the number of hot spots from 6 to 10 the percentage performance increase is about 35% independently of the number of used nanosatellites, as evidenced by the vertical ellipses in Fig. 8(b), again for 4, 10, and 18 nanosatellites. Equivalently, augmenting the number of nanosatellites, the percentage variation is almost constant independently on the hot spot number: increasing the number of nanosatellites from 6 to 8 the percentage performance increase by using 1 and 6 hot spots, respectively, is very similar (21% and 25%), as evidenced in Figs. 8(a) and 8(b) through horizontal ellipses. The same behaviour may be detected by changing the number of nanosatellites from 14 to 16 in the same figures.

D. Quality of Service Versus Quality of Experience for Possible Internet Web Browsing

In order to exploit the increase of transmission rate, number of nanosatellites, hotspots, and cold spots and to check the feasibility of regular web browsing through HotSel, we show the obtained performance results in terms of Mean Opinion Score (MOS). In this way, we provide a view of the performance advantages not only in terms of an objective Quality of Service (QoS) parameter (ADT), but also in terms of Quality of Experience (QoE). The results shown in the previous paragraphs only consider the one way delivery time on the backward path (from the Internet to the rural areas). The time values considered for the MOS results also include an estimation of the time required on the forward path (from the Internet to the rural areas) to deliver the rural user requests. This estimation depends on the number of satellites and ground stations in the network and on the consideration that the request size is much smaller than the response one. As a consequence, all waiting requests in the cold spot buffer will be uploaded on the first satellite in contact and will be downloaded to the first hot spot along the satellite path. The Complete Delivery Time (CDT) used as input to the QoE model is defined in (19).

---

[2]Multiple orbital planes are not considered in this paper. Increasing the number of nanosatellites means having more satellites within the same orbit.
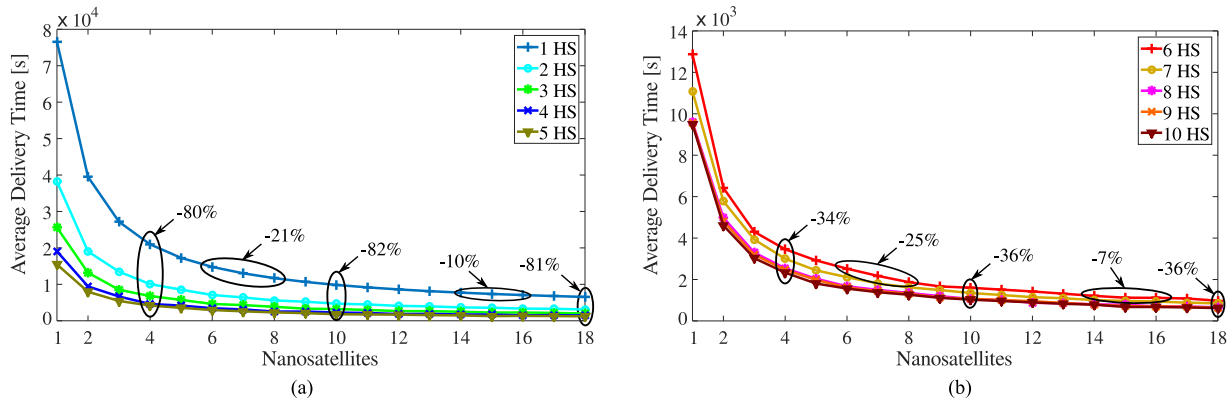
$$CDT = ADT + 2 * \frac{W}{N_{SAT}} + \frac{W}{N_{HS}} \qquad (19)$$

Fig. 8.   ADT varying the nanosatellites number from 1 to 18 and hot spots number from 1 to 5 (a) and from 6 to 10 (b).

where W, N_{SAT}, and N_{HS}, as defined in Section V, are, respectively, the orbit time, the number of nanosatellites, and the number of hot spots.

As described in [43], there are several models to map user waiting times over the QoE for web services. The first model (MOS-L) we use is based on the WQL hypothesis [44] where the relation QoE-QoS is given by the Weber-Fechner Law (WFL) [45]:

$$MOS(CDT) = \alpha * \ln(CDT) + \beta \qquad (20)$$

The second model is based on the IQX hypothesis [46]. The formula which describes the QoE-QoS relation is:

$$MOS(CDT) = \alpha * e^{-\gamma * CDT} + \beta \qquad (21)$$

Values α and β in (20) can be computed, as indicated in [47], by setting the minimum ($CDT_{\min}$) and maximum ($CDT_{\max}$) CDT values respectively corresponding to $MOS = 5$ and $MOS = 0$. In practice, solving the system composed of the two equations: $0 = \alpha * \ln(CDT_{\min}) + \beta$; $5 = \alpha * \ln(CDT_{\max}) + \beta$. The procedure to get α, β, and γ in (21) is the same.

Since we are using a sort of delay-tolerant web browsing (i.e., a web service where large delays may be accepted), we can assume that the users in rural areas are aware that the offered service cannot be as quick as a typical Internet web browsing. Consequently we set $CDT_{\min} = 50$ s and $CDT_{\max} = 1500$ s, getting $\alpha = -1.47$ and $\beta = 10.75$ for the WQL hypothesis and $\alpha = 5.85$, $\beta = -0.29$, and $\gamma = 0.002$ for the IQX hypothesis.

Even in the delay-tolerant context, the obtained ADT values in Scenarios 1 and 2 are too large to successfully enforce the chosen models which are defined for a typical web browsing service. To deal with this, we have performed simulations increasing further the number of ground stations and nanosatellites in order to obtain CDT values somehow compatible with a typical web browsing and to apply MOS models. The two simulated scenarios for the following tests are composed of 100 hot spots, 500 nanosatellites, 400 cold spots (Scenario 3) and of 110 hot spots, 550 nanosatellites, 440 cold spots (Scenario 4). All traffic flows have a uniform traffic generation function (1000 bundles, 1 per second). The obtained CDT values are shown in Table IX. $4CP$ (four consecutive orbit portions) and $4NCP$ (four not consecutive orbit portions) are two new traffic flow configurations defined conformally with the definitions in

TABLE IX
CDT FOR SCENARIOS 3 AND 4

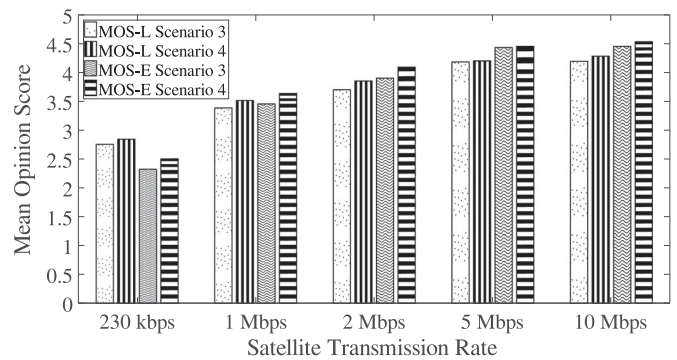|		1P	2CP	2NCP	4CP	4NCP
230 kbps	**Sc. 3**	299	398	301	546	304
	Sc. 4	274	365	272	513	276
1 Mbps	**Sc. 3**	231	243	235	284	235
	Sc. 4	197	224	196	260	197
2 Mbps	**Sc. 3**	163	178	167	192	166
	Sc. 4	143	157	140	174	145
5 Mbps	**Sc. 3**	109	112	109	117	109
	Sc. 4	105	110	103	120	106
10 Mbps	**Sc. 3**	108	109	108	113	108
	Sc. 4	98	101	98	105	98

Fig. 9.   Mean opinion score for Scenarios 3 and 4.

Section VIII-A. Fig. 9 shows the obtained MOS by using (20) and (21) after averaging the CDT values in Table IX over the 5 used traffic flow configurations.

Even if CDT values in Table IX are not comparable with transfer delays measured for regular web navigation, the results obtained for 5 and 10 Mbps, whose corresponding MOS ranges between 4.17 and 4.52, show that the proposed solution can offer a delay-tolerant web navigation assuming a minimum interaction and able to satisfy rural users.

E. Nanosatellite Buffer Occupancy

For the same set of simulations, we have also considered the trend of the nanosatellite buffer occupancy, in order to analyse its
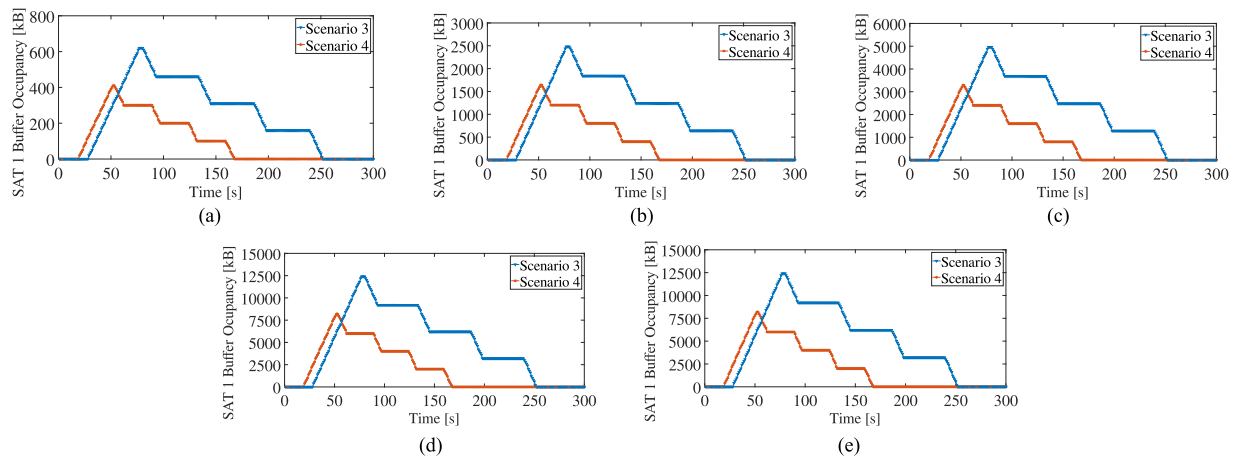
Fig. 10.　SAT_1 buffer occupancy trend by varying satellite transmission rate for Scenarios 3 and 4. (a) 230 kbps, (b) 1 Mbps, (c) 2 Mbps, (d) 5 Mbps, (e) 10 Mbps.
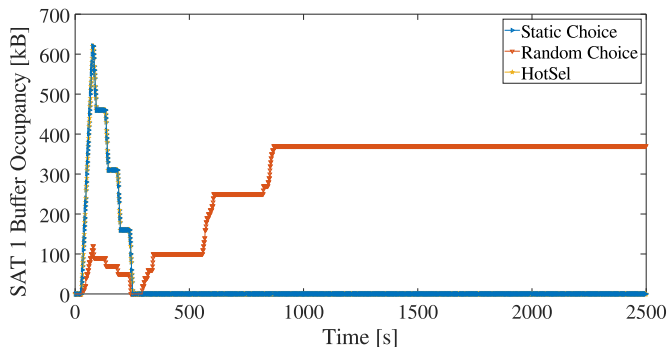

Fig. 11.　SAT_1 buffer occupancy trend by varying the hot spot selection mechanism in Scenario 3.

relation with the satellite transmission rate and the implemented network load configurations. Fig. 10 shows the buffer occupancy trend of one satellite ($SAT1$ has been chosen in this case) obtained in one network load configuration ($1P$) for Scenarios 3 and 4 changing the satellite transmission rate.

The buffer occupancy trend of the other nanosatellites follows the same behaviour shown in Fig. 10 for all cases and it is not reported in the paper. The curves in Fig. 10 have the same behaviour: nanosatellites upload bundles when they enter in contact with a hot spot, so nanosatellite buffer occupancies grow up; nanosatellites download bundles when they enter in contact with their destination cold spots and, consequently, buffer occupancies decrease. This last action happens 4 times because, in the simulated Scenarios 3 and 4, 4 cold spots are positioned between each couple of consecutive hot spots. The different numerical values appearing in Fig. 10 for the two scenarios are justified by the contact duration and by the distance between ground stations, both lower in Scenario 4. Showing the buffer occupancy over time is also helpful to understand the behaviour of the different hot spot selection mechanism. Fig. 11 shows how SAT_1 buffer occupancy changes by varying the hot spot selection mechanisms.

The comparison is limited to HotSel and Random Choice because, in this case, the behaviour of HotSel and Static Choice is totally overlapped. The reason is that increasing the number of

nanosatellites, the number of contacts augments, and, again, as expressed in Section VIII-B concerning the transmission rate, the "nearest" hot spot to the destination cold spot is congestion free and is chosen by HotSel. HotSel allows avoiding that nanosatellites have to store bundles for long time before downloading them. Consequently, the risk of losses due to buffer overflow is reduced respect to the Random Choice, as evident also from Fig. 11.

F. Network Performance Versus Costs

From the practical viewpoint it is important to evaluate the trade-off between network performance and costs.

The employment and operating cost of a network with a given number of hot spots (N_{HS}), cold spots (N_{CS}, as defined in Section V), and nanosatellites (N_{SAT}) is computed as:

$$C = (N_{HS} + N_{CS}) \cdot C_{GS} + N_{SAT} \cdot C_{A,SAT}$$
$$+ \left\lceil \frac{N_{SAT}}{3} \right\rceil \cdot C_{L,3SAT} + C_{Opy} \cdot L_y \qquad (22)$$

C_{GS} is the estimated cost per ground station; $C_{A,SAT}$ is the estimated assembly cost per nanosatellite; $C_{L,3SAT}$ is the estimated launch cost for a group of three nanosatellites; C_{Opy} is the operating cost per year and L_y is the network lifetime. Referring to [48], these costs are: $C_{GS} = 2\,000$ \$, $C_{A,SAT} = 100\,000$ \$, $C_{L,3SAT} = 200\,000$ \$, and $C_{Opy} = 1\,000\,000$ \$ with a network lifetime of 5 years. [48] considers an uplink/downlink transmission rate of 230 kbps. All user cost parameters are referred to it. Changing the transmission rate implies changing the mentioned cost values and the numerical results reported in the remainder of the paper.

In this way, it is possible to compute how much money is needed to invest in order to deploy and maintain a nanosatellite-DTN network also depending on the level of performance provided to rural users in terms of delivery time.

We have simulated a set of scenarios with a fixed number of rural areas and traffic flows (both set to 8), an increasing number of hot spots (from 1 to 10), and an increasing number of nanosatellites (from 1 to 18), conformally with Fig. 8. Fig. 12
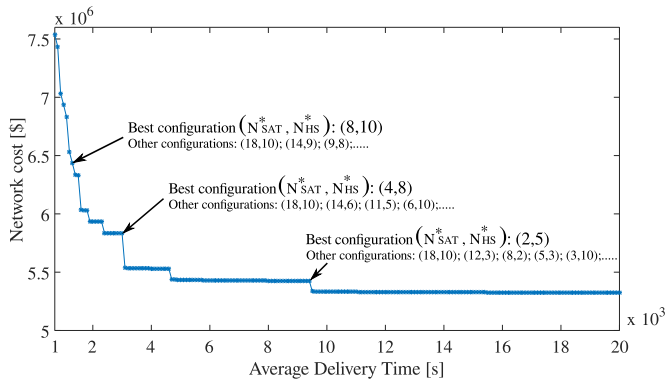
Fig. 12.   Relationship between ADT and network cost.

shows the cost related to the optimal network configuration necessary to keep the delivery time under a chosen threshold. The optimal configuration is computed as the minimum number of nanosatellites N^*_{SAT} and hot spots N^*_{HS} so that C in (22) is minimized:

$$(N^*_{SAT}, N^*_{HS}) = \operatorname*{arg\,min}_{(N_{SAT}, N_{HS})} C. \qquad (23)$$

A practical impact: the overall cost to get a delivery time of 8 000 s is \$5.4 million, assured by the optimal configuration (2,5), as shown in Fig. 12. A delivery time of about 2 500 s requires about \$5.8 million, and a delivery time of about 700 s requires about \$7.5 million. The cost needed to guarantee lower delivery times such as the ones obtained for Scenarios 3 and (corresponding CDT values are shown in the first two lines of Table IX) 4 are of about \$90 and \$98 million, respectively.

IX. CONCLUSION

In this paper we have presented "gRANteD": a novel Nanosatellite-DTN architecture aimed at extending network access to rural and remote areas ensuring a transparent communication among TCP/IP endpoints.

gRANteD includes "HotSel", a hot spot selection algorithm that minimizes the delivery time of data in a nanosatellite-DTN network.

The obtained results show that, when there are congestion situations in hot spots, HotSel reduces the bundle delivery time in all the simulated scenarios and traffic flow configurations with respect to two other test mechanisms: static and random selection. The results show that more the destination rural nodes are concentrated in adjacent areas, higher the performance improvement. It is possible to obtain different levels of QoS performance depending on the employed number of ground stations and nanosatellites and on the uplink/downlink transmission rate. In particular, using higher transmission rates together with a large number of hot spots, nanosatellites, and cold spots (as done in Scenarios 3 and 4) allows getting Complete Delivery Time values fully compatible with a delay-tolerant web navigation. To this aim, we have also shown the relation between the obtained delivery time and the Mean Opinion Score estimating the user satisfaction. The relation between the obtained

performances and the network cost in order to know how much money it is necessary to invest to build a network guaranteeing a delivery time under a fixed and chosen threshold is also discussed.

REFERENCES

[1] Internet usage statistics. [Online]. Available: http://www.internetworldstats.com/stats.htm

[2] Y. Kawamoto, H. Nishiyama, N. Kato, and N. Kadowaki, "A traffic distribution technique to minimize packet delivery delay in multilayered satellite networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3315–3324, Sep. 2013.

[3] S. Burleigh, "Nanosatellites for universal network access," in *Proc. ACM MobiCom Workshop Lowest Cost Denominator Netw. Universal Access*, 2013, pp. 33–34.

[4] H. Heidt, J. Puig-Suari , A. Moore, S. Nakasuka, and R. J. Twiggs, "Cubesat: A new generation of picosatellite for education and industry low-cost space experimentation," in *Proc. 14th Annu./USU Conf. Small Satellites*, Aug. 2000, pp. 1–19.

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

[6] V. Cerf *et al.*, "Delay-tolerant networking architecture," RFC4838, Apr. 2007.

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

[8] A. Doria, M. Uden, and D. P. Pandey, "Providing connectivity to the saami nomadic community," in *Proc. 2nd Int. Conf. Open Collaborative Des. Sustain. Innov.*, Dec. 2002, pp. 1–8.

[9] A. S. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, Jan. 2004.

[10] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, 2006, pp. 334–345.

[11] B. Raman and K. Chebrolu, "Experiences in using wifi for rural internet in India," *IEEE Commun. Mag.*, vol. 45, no. 1, pp. 104–110, Jan. 2007.

[12] P. Manzoni, C. T. Calafate, J.-C. Cano, and E. Mota, "Epidgeons: Combining drones and dtns technologies to provide connectivity in remote areas," in *Proc. 1st Workshop Micro Aerial Vehicle Netw. Syst. Appl. Civilian Use*, 2015, pp. 57–58.

[13] A. Galati, T. Bourchas, S. Siby, and S. Mangold, "System architecture for delay tolerant media distribution for rural South Africa," in *Proc. 9th ACM Int. Workshop Wireless Netw. Testbeds Exp. Eval. Characterization*, 2014, pp. 65–72.

[14] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: Realizing multihop device-to-device communications," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 56–65, Apr. 2014.

[15] A. Mori, H. Okada, K. Kobayashi, M. Katayama, and K. Mase, "Construction of a node-combined wireless network for large-scale disasters," in *Proc. 12th Annu. IEEE Consumer Commun. Netw. Conf.*, 2015, pp. 219–224.

[16] N. Uchida, N. Kawamura, and Y. Shibata, "Evaluation of cognitive wireless based delay tolerant network for disaster information system in a rural area," in *Proc. 7th Int. Conf. Complex Intell. Softw. Intensive Syst.*, 2013, pp. 1–7.

[17] C.-W. Lin *et al.*, "Empowering village doctors and enhancing rural healthcare using cloud computing in a rural area of mainland china," *Comput. Methods Programs Biomed.*, vol. 113, no. 2, pp. 585–592, 2014.

[18] M. Barua, X. Liang, R. Lu, and X. S. Shen, "Rcare: Extending secure health care to rural area using vanets," *Mobile Netw. Appl.*, vol. 19, no. 3, pp. 318–330, 2014.

[19] GAIA, "Global access to the internet for all research group." [Online]. Available: https://irtf.org/gaia

[20] S. Rushing, "Analyzing "global access to the internet for all" projects," Network Architectures and Services, Seminars FI / IITM WS 15/16, 2016.

[21] Google, Project loon. [Online]. Available: http://www.google.com/loon/

[22] Facebook and partners, Internet.org. [Online]. Available: http://internet.org/

[23] RIFE, "Architecture for an internet for everybody project." [Online]. Available: https://rife-project.eu/about/

[24] ABSOLUTE, Aerial base stations with opportunistic links for unexpected and temporary events project. [Online]. Available: http://www.absolute-project.eu/

[25] Spacex. [Online]. Available: http://www.spacex.com/

[26] Oneweb. [Online]. Available: http://www.oneweb.world/

[27] Iridium, "Iridium global network," 1998. [Online]. Available: http://www.iridium.com/About/IridiumGlobalNetwork.aspx

[28] Globalstar, "Globalstar network," 1991. [Online]. Available: http://eu.globalstar.com/en/index.php?cid=3300

[29] Orbcomm, "Orbcomm networks," 1991. [Online]. Available: http://www.orbcomm.com/networks

[30] Iridium, "Iridium next." [Online]. Available: https://www.iridium.com/network/iridiumnext

[31] Inmarsat, "Inmarsat satellites," 1990. [Online]. Available: http://www.inmarsat.com/about-us/our-satellites

[32] S. C. Burleigh, "Contact graph routing," Internet-Draft, 2009. [Online]. Available: https://www.ietf.org/archive/id/draft-burleigh-dtnrg-cgr-01.txt

[33] I. Bisio and M. Marchese, "Satellite earth station (ses) selection method for satellite-based sensor networks," *IEEE Commun. Lett.*, vol. 11, no. 12, pp. 970–972, Dec. 2007.

[34] M. Cello, M. Marchese, and F. Patrone, "Hot spot selection in rural access nanosatellite networks," in *Proc. 9th ACM MobiCom Workshop Challenged Netw.*, 2014, pp. 69–72.

[35] M. Cello, M. Marchese, and F. Patrone, "Hotsel: A hot spot selection algorithm for internet access in rural areas through nanosatellite networks," in *Proc. Global Commun. Conf.*, 2015, pp. 1–6.

[36] M. Cello, M. Marchese, and F. Patrone, "Satsel: A satellite selection algorithm to reduce delivery time in dtn-nanosatellite networks for internet access in rural areas," in *Proc. 8th Adv. Satellite Multimedia Syst. Conf. 14th Signal Process. Space Commun. Workshop*, 2016, pp. 1–6.

[37] M. Cello, M. Marchese, and F. Patrone, "Coldsel: A selection algorithm to mitigate congestion situations over nanosatellite networks," in *Proc. IEEE Global Commun. Conf.*, 2016, pp. 1–6.

[38] K. L. Kusza and M. A. Paluszek, "Intersatellite links: Lower layer protocols for autonomous constellations," in First Joint Space Internet Workshop, NASA Goddard Space Flight Center, Maryland, USA, 2000.

[39] A. Houyou, R. Holzer, H. Meer, and M. Heindl, "Performance of transport layer protocols in leo pico-satellite constellations," Univ. Passau, Passau, Germany, Tech. Rep. MIP-0502, 2005.

[40] C. Caini *et al.*, "Transport layer protocols and architectures for satellite networks," *Int. J. Satellite Commun. Netw.*, vol. 25, no. 1, pp. 1–26, 2007.

[41] J. Lakkakorpi, "Dtn code for ns-2 and ns-3," 2013. [Online]. Available: http://www.netlab.tkk.fi/tutkimus/dtn/ns/

[42] G. Araniti *et al.*, "Contact graph routing in dtn space networks: Overview, enhancements and performance," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 38–46, Mar. 2015.

[43] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler, "Waiting times in quality of experience for web based services," in *Proc. 4th Int. Workshop Qual. Multimedia Experience*, 2012, pp. 86–96.

[44] S. Egger, P. Reichl, T. Hosfeld, and R. Schatz, "Time is bandwidth? Narrowing the gap between subjective time perception and quality of experience," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 1325–1330.

[45] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo, "The logarithmic nature of qoe and the role of the weber-fechner law in qoe assessment," in *Proc. IEEE Int. Conf. Commun.*, 2010, pp. 1–5.

[46] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Netw.*, vol. 24, no. 2, pp. 36–41, Mar./Apr. 2010.

[47] "Estimating end-to-end performance in ip networks for data applications," Int. Telecommun. Union, Geneva, Switzerland, ITU-T Rec G.1030, Nov. 2005.

[48] S. C. Burleigh, "Ring road: User application on a high-latency network," Jan. 2014. [Online]. Available: http://ipnsig.org/ipn-events/2014-space-technology-innovations/

Mario Marchese (S'94–M'97–SM'04) was born in Genoa, Italy, in 1967. He received the "Laurea" degree cum laude and the Ph.D. (Italian "Dottorato di Ricerca") degree in "Telecommunications" from the University of Genoa, Genoa, Italy, in 1992 and 1997, respectively.

From 1999 to January 2005, he worked with the Italian Consortium of Telecommunications (CNIT), by the University of Genoa Research Unit, where he was the Head of Research. From February 2005 to January 2016, he was an Associate Professor at the University of Genoa, where he has been a Full Professor since February 2016. He is the author of the book *Quality of Service over Heterogeneous Networks* (John Wiley & Sons, Chichester, 2007) and author/co-author of more than 290 scientific works, including international magazines, international conferences, and book chapters. His main research interests include networking, quality of service over heterogeneous networks, software defined networking, satellite dtn and nanosatellite networks, networking security.

He was the Chair of the IEEE Satellite and Space Communications Technical Committee from 2006 to 2008. He received the IEEE ComSoc Award "2008 Satellite Communications Distinguished Service Award" in "recognition of significant professional standing and contributions in the field of satellite communications technology."

Fabio Patrone was born in Genoa, Italy, in 1988. He received the Bachelor's degree in telecommunication engineering and the Master's degree in telecommunication engineering in 2013 both from the University of Genoa, Genoa, and the Ph.D. degree from the Satellite Communications and Networking Laboratory (SCNL), University of Genoa, with a thesis on routing and scheduling algorithms in Satellite Delay and Disruption Tolerant Networks (DTNs). He is currently a Postdoctoral Research Fellow at the SCNL. His main research interests include satellite networks and DTN networks, in particular design of routing, scheduling, and congestion control algorithms for satellite networks.

Marco Cello received the "Laurea Magistrale" (M.Sc.) degree cum laude and the Ph.D. degree both from the University of Genoa, Genoa, Italy, in 2008 and 2012, respectively. In 2012, 2014, and 2015, he worked as Postdoctoral Research Fellow with the University of Genoa with a fellowship funded by Fondazione Carige. In 2013, he was Postdoctoral Research Fellow at Polytechnic Institute of New York University and Visiting Research Fellow at New York University Abu Dhabi. He is currently a Postdoc Research Fellow at Nokia Bell Labs in Dublin, Ireland. He is a Researcher on networking with almost 8 years of experience, including managing research projects funded by national industries, the European Community and the European Space Agency. He has strong expertise in software for simulation, Linux-based emulation of telecommunication networks and Linux administration. He is a co-author of more than 20 scientific works, including international journals, conferences, and patents. His main research interests include online modeling and understanding of carrier-grade VM-based cloud services and software-defined networks. He has also interests in network modeling/teletraffic engineering and delay-tolerant networks.