SatSel: A Satellite Selection Algorithm to reduce delivery time in DTN-Nanosatellite Networks for Internet Access in Rural Areas

Marco Cello[† §], Mario Marchese[§], Fabio Patrone[§]

[†] Nokia Bell Labs, Dublin, Ireland
[§] University of Genoa, Genoa, Italy

Email: marco.cello@nokia.com, mario.marchese@unige.it, f.patrone@edu.unige.it

***Abstract*—There are some different ways to connect rural areas to the Internet. One of these provides the use of a nanosatellite constellation. This type of network allows people in rural areas to enjoy all services the Internet can offer keeping low the cost of Internet access. One of the critical aspect is related to the delivery time, because LEO satellite links are not always up. This means that the system must be able to deal with periodic disruptions and high delays in the path from the source to the destination, considering that data could be stored in nanosatellite, Internet gateway (also called hot spot), and rural gateway (also called cold spot) buffers also for several seconds or minutes waiting to be forwarded. In the path from rural areas to the Internet, it is possible to reduce data delivery time acting on rural gateways. We propose SatSel: a selection algorithm which allows the cold spots to choose the nanosatellite to whom upload data in order to reduce the data delivery time.**

I. Introduction

The problem of Internet access in rural areas becomes more and more attractive day by day. Some projects were started in recent few years with the final goal of providing Internet access to all people of the world, especially people do not have it yet. Google's Project Loon [7] involves the use of a network of balloons travelling at an altitude of about 20 km, Facebook and partners' project Internet.org [6] provides the use of drones, SpaceX and partners' project [11] and the Oneweb project [9] are based on the utilization of a huge number of microsatellites, a type of Low Earth Orbit (LEO) satellites.

Other already proposed solutions provides the use of Nanosatellites [1] as a cost-effective solution to extend Internet access in rural and remote areas. Rural and/or disconnected areas will be connected through local gateways that will communicate in an opportunistic fashion with the nanosatellite constellation using the Delay Tolerant Networking (DTN) paradigm [2], [5].

Figure 1 shows a DTN-Nanosatellite network scenario: in a rural area, a group of users or nodes $R_1, \ldots, R_N$ is connected with the node CS_1. Nodes CS_1 and CS_2, referred in the following as "cold spots" (CSs), are located in remote areas and act as Internet gateways for rural users. Nanosatellites SAT_1, SAT_2, and SAT_3 upload and download data from nodes CS_1 and CS_2 but also from nodes HS_1 and HS_2, referred in the following as "hot spots" (HSs), which are linked to the Internet. Hot spots send the requests to the central node C of the constellation that opens the communication with servers on the Internet (e.g. node D).

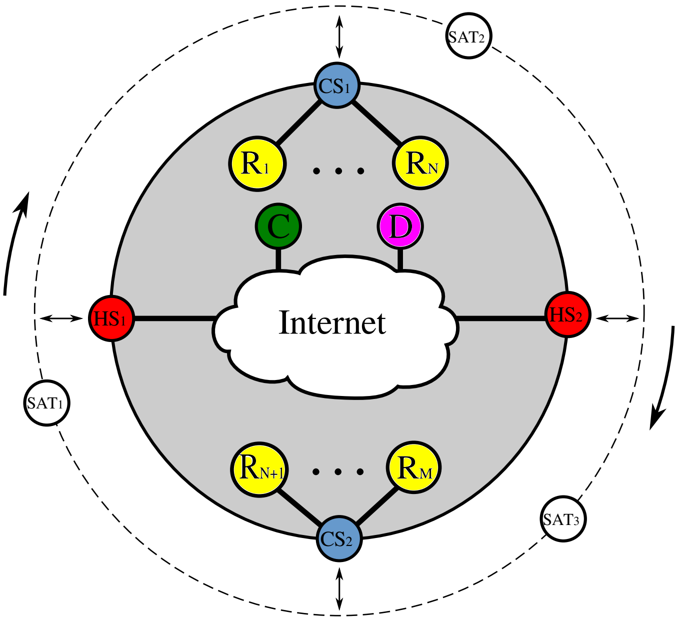


Figure 1: Nanosatellite network scenario.

When a rural user wants to gather information (such as a web page) from the Internet, or has some data to send to an Internet Server (such as a Mail Server or a Cloud Server), it sends its data to the cold spot that manages the remote region where it is located. The cold spot waits until a nanosatellite comes in contact with it and uploads the data. When the nanosatellite comes in contact with the first available hot spot on its route, it forwards the information to that hot spot which forwards again to the central node C. C reads the message to know which is the destination node (e.g. D) and delivers all packets to it.

Since nanosatellites are a type of LEO satellites, there cannot be a persistent path between source and destination. The DTN architecture provides long term information storage on 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)

[10], 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”. It is a message that encapsulates application layer protocol data units. The DTN paradigm and BP are employed just to cope with this problem. Moreover, a limited amount of data can be exchanged between nanosatellites and ground stations during each contact, due to their limited duration. Choose the right contact between nanosatellites and cold spots reduces data delivery time for all traffic from rural areas to the Internet, which is a critical performance parameter for some services (VoIP, gaming, ...). In this article we propose “SatSel”: a selection algorithm which allows the cold spots to choose to which nanosatellite upload data destined to endpoints on the Internet, in order to reduce data delivery time of traffic flows from rural areas to the Internet. This algorithm takes advantage of the same principles applied in [3] to develop an hot spot selection algorithm (called HotSel) to reduce delivery time for all data on the reverse path, from the Internet to the rural areas.

II. Nanosatellite Selection

In this section we describe “SatSel”: a dynamic nanosatellite selection method implemented in all cold spots. The purpose of SatSel is to reduce the delivery time of all data bundles destined to the Internet servers, allowing each cold spot to choose to upload each bundle on the nanosatellite which it is in contact with or to keep the bundle stored in its buffer in order to subsequently upload it on another nanosatellite. To do this, cold spots need to know some information about network topology, nanosatellite buffer occupancies, and contacts between ground stations and nanosatellites. Some of these information do not change (ground stations number and position are fixed, nanosatellite number is fixed, nanosatellite position changes in a deterministic way), but some others change during network lifetime (number of bundles stored in nanosatellites and ground stations buffer). Information about the former set can be stored in all network nodes during the installation phase, but information about the latter set must be periodically updated to avoid cold spots make the wrong choice. However, since cold spots are located in rural areas, they can collect these information only from the nanosatellites when they will come in contact with each other. During each contact, the nanosatellite sends a bundle to make the cold spot aware of its buffer situation, but it does not have any information about buffer situation of the other nanosatellites. Different from the algorithm proposed in [3], this algorithm could not always make the best choice, because some of the needed information are estimated.

When a bundle b needs to be transmitted from CS_P to the central node C, SatSel computes the delivery time T_k needed to transmit b through SAT_k as:

$$T_k = w_{P,k} + t_{P,k}, \tag{1}$$

where $w_{P,k}$ is the flight time of SAT_k from its current position to its contact with CS_P and $t_{P,k}$ is the flight time of SAT_k between its contact with CS_P and its contact with the first hot spot able to receive b. SatSel iterates on all nanosatellites.

The SAT_K that minimizes the delivery time for bundle b is:

$$K = \underset{k \in \mathcal{SAT}}{\operatorname{argmin}}\, T_k, \tag{2}$$

where $\mathcal{SAT}$ is the set of nanosatellites.

In this way, when CS_P enters in contact with a SAT_k, it can decide if it is better to upload b on that nanosatellite or wait until it will come in contact with another one.

If CS_P is in contact with SAT_K, $w_{P,K} = 0$, and the other values of $w_{P,k}$ can be calculated as:

$$w_{P,k} = S_k * \frac{T_{ORB}}{N_{SAT}}, \tag{3}$$

where S_k is the number of nanosatellites that will enter in contact with CS_P before SAT_k, T_{ORB} is the orbit time of nanosatellites, and N_{SAT} is the number of nanosatellites in the network (so T_{ORB}/N_{SAT} is the average flight time between two consecutive nanosatellites). For example, referring to the scenario in Figure 1, if SAT_1 is in contact with CS_1, $S_1 = 0$, $S_2 = 1$, and $S_3 = 2$.

Assuming all nanosatellites located on one circular orbit, all contacts between ground stations and nanosatellites have the same duration and can be exchanged the same maximum amount of data. In this way, to calculate the time $t_{P,k}$, it is necessary to define the following parameter:

$$N^k_{CONT} = \left\lceil \frac{B^k_I + B}{Q} \right\rceil, \tag{4}$$

where B^k_I is the amount of data already stored in the buffer of SAT_k and destined to the Internet, B is the size of bundle b, and Q is the amount of data that each nanosatellite can upload/download during each contact. N^k_{CONT} indicates the number of contacts between SAT_k and the hot spots necessary to download all data stored in SAT_k and destined to the Internet, included bundle b. We can compute $t_{P,k}$ as:

$$t_{P,k} = \begin{cases} T_{CS_P \to HS_J} & N^k_{CONT} = 1, \\ T_{CS_P \to HS_J} + \sum_{j=1}^{N^k_{CONT}-1} T_{HS_{J+j} \to HS_{J+j+1}} & otherwise, \end{cases} \tag{5}$$

where $T_{CS_P \to HS_J}$ is the flight time of each nanosatellite between its contact with CS_P and its contact with the first hot spot after CS_P on the nanosatellite path (HS_J) and $T_{HS_{J+j} \to HS_{J+j+1}}$ is the flight time of each nanosatellite between two consecutive hot spots after HS_J on the nanosatellite path.

The time $t_{P,k}$ depends on the buffer occupancy of SAT_k, because which will be the first hot spot able to receive bundle b depends on the amount of data destined to the Internet already stored in the buffer of SAT_k (B^k_I). If SAT_k is in contact with CS_P, it can directly make the cold spot aware of this information using an ad-hoc protocol, but the information about the buffer occupancies of the other nanosatellites are not available, because they are far from CS_P and they cannot communicate with it. It is so necessary to estimate the amount of data destined to the Internet stored in all nanosatellite buffers when they will come in contact with CS_P. To do this, there are two possible ways:

- Use some traffic models to estimate the amount of data generated by users located in each rural area and destined to the Internet. This is achievable considering the knowledge of the number and position of rural areas and ground stations in the network, and it requires the estimation of the mean number of rural users for each rural area.

- Allow each cold spot to store the buffer occupancy informations sent by nanosatellites in the occurred contacts with each other, in order to create a nanosatellite buffer occupancy history.

The first solution is more complex than the latter. It is not easy to estimate the number of rural users in each rural areas, which depends on different parameters (such as area extension, population density, ...) and will change during network lifetime. Moreover, not all users are the same: there could be both citizens and tourists, which could generate different types and amounts of traffic. Instead, the second solution requires a learning phase necessary to create the nanosatellite buffer occupancy history.
We chose to further investigate and test the algorithm using the second proposal. In this way, each cold spot can estimate B_I^k as the mean of the nanosatellite buffer occupancy values collected during its previous contacts:

$$\tilde{B}_I^k = \begin{cases} 0 & N_{CONT}^P = 0, \\ \frac{\sum_{t=1}^{N_{CONT}^P} B_I(t)}{N_{CONT}^P} & otherwise, \end{cases} \tag{6}$$

where N_{CONT}^P is the number of occurred contacts between CS_P and nanosatellites and $B_I(t)$ is the amount of data destined to the Internet stored in the buffer of the nanosatellite that came in contact with CS_P in its t^{th} contact from the start time of the network. This estimation of B_I^k is strong against single anomalous values, which can be due to power failures (quite frequent in rural areas of underdeveloped countries [8]), even though it suffers due to the need of a learning phase. Moreover, it is possible to opportunely set a maximum number of previous values to consider in the mean computation, in order to not take into account too old values, or define a weights set, in order to give greater relevance to the most recent values.

III. Performance Analysis

We implemented SatSel in our DTN module for the software Network Simulator 3 (NS3) [4].
We performed a set of simulations by using four different scenarios, changing the number of ground stations and nanosatellites:

1) **Scenario 1**: it is composed of 2 hot spots (HS_1 and HS_2), 4 nanosatellites (SAT_1-SAT_4), 12 cold spots (CS_1-CS_{12}) 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, ...).
2) **Scenario 2**: it is composed of 2 hot spots (HS_1 and HS_2), 8 nanosatellites (SAT_1-SAT_8), 12 cold spots (CS_1-CS_{12}) and 2 rural nodes for each cold spot (R_1-R_{24}).
3) **Scenario 3**: it is composed of 4 hot spots (HS_1-HS_4), 4 nanosatellites (SAT_1-SAT_4), 16 cold spots (CS_1-CS_{16}) and 2 rural nodes for each cold spot (R_1-R_{32}).
4) **Scenario 4**: it is 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}).

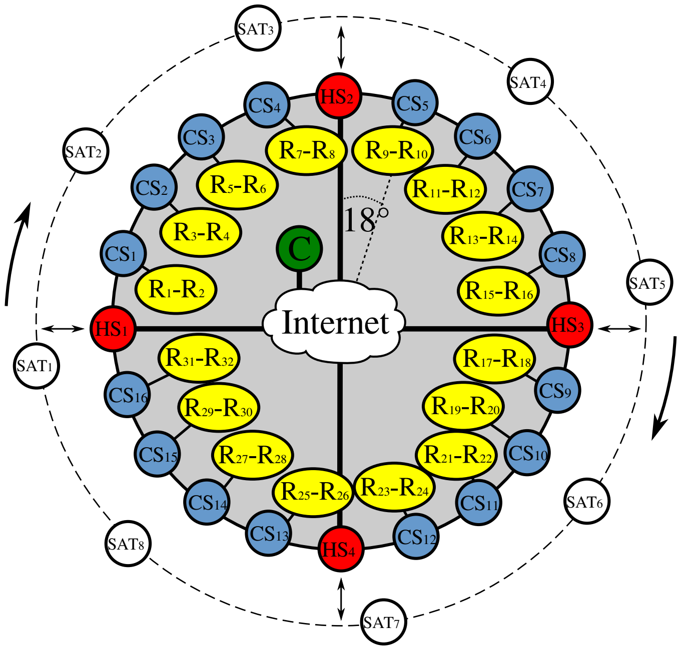


Figure 2: Scenario 4

Figure 2 shows one of the simulated scenarios (Scenario 4).

In all scenarios, all ground stations (both hot spots and cold spots) are equally spaced, also the distance between two consecutive nanosatellites is constant, since they are located in the same circular orbit. We assume that the nanosatellites keep the same speed even though in a real scenario is not exactly so. The nanosatellites altitude is 200 km, consequently the orbit time is about 90 minutes and the contact time is about 256 s. The transmission rate of satellite links is 230 Kbps, so ground stations can upload on nanosatellites and nanosatellites can download to ground stations about 7 MB of data in each contact (satellite links are full duplex). This amount is an underestimation, because we have set margin times at the beginning of contacts to exchange information about nanosatellite buffer occupancies. All traffic flows have a fixed number of bundles with the same size M.
Considering all these assumptions, equations (4) and (5) can be rewritten as:

$$N_{CONT}^k = \left\lceil \frac{M * (N_{B_I^k} + 1)}{Q_B} \right\rceil, \tag{7}$$

$$t_{P,k} = T_{CS_P \rightarrow HS_J} + N_{CONT}^k * \frac{T_{ORB}}{N_{HS}}, \tag{8}$$

where $N_{B_I^k}$ is the number of bundles already stored in the buffer of SAT_k and destined to the Internet, Q_B is the number of bundles that each nanosatellite can upload/download during each contact, and N_{HS} is the number of hot spots in the network (so T_{ORB}/N_{HS} is the average flight time between two consecutive hot spots).
Each cold spot can estimate $N_{B_I^k}$ as the mean of the buffer occupancy values obtained during its previous contacts with

SAT_k. Also equation (6) can be rewritten as:

$$\tilde{N}_{B_I^k} = \begin{cases} 0 & N_{CONT}^{k \leftrightarrow P} = 0, \\ \frac{\sum_{t=1}^{N_{CONT}^{k \leftrightarrow P}} N_{B_I^k}(t)}{N_{CONT}^{k \leftrightarrow P}} & otherwise, \end{cases} \quad (9)$$

where $N_{CONT}^{k \leftrightarrow P}$ is the number of occurred contacts between SAT_k and CS_P we want to consider and $N_{B_I^k}(t)$ is the number of bundles destined to the Internet stored in the buffer of SAT_k when it came in contact with CS_P in their t-th contact of the contact set we decide to consider.
The topology of the simulated scenarios are based on the "Ring Road" concept mentioned in [1]. We define as orbit portion each of the four equal parts obtained dividing the orbit footprint in a clockwise direction starting from HS_1. These four portions are called north-east, north-west, south-east, and south-west portions. For example, looking at Figure 2, in Scenario 4 the north-west portion includes the orbit footprint from HS_1 to HS_2, the north-east portion from HS_2 to HS_3, the south-east portion from HS_3 to HS_4, and the south-west portion from HS_4 to HS_1.
For each scenario, we performed different simulations changing the network load configuration, i.e. the number and position of source nodes, in order to test the performance in different realistic network load situations. However, since the purpose of SatSel is to reduce the delivery time of bundles from rural areas to the Internet, all source nodes are always located in rural areas and the destination node is always the central node C.
The links between rural users and cold spots are wired links, even though can also be simulated using wireless links without any changes in the obtained results. The following formalism has been used: the notation R_x-R_y-R_z-R_t means that the source nodes are the rural nodes R_x, R_y, R_z, and R_t at the same time. To better quantify the performance improvement achievable by using SatSel, we decided to test four different network load configurations:

1) **One portion (1P)**: all traffic flow source nodes are located in the same orbit portion (north-west portion). It regards the configurations named R_1-R_3-R_5 for Scenario 1 and Scenario 2, and R_1-R_3-R_5-R_7 for Scenario 3 and Scenario 4.
2) **Two consecutive portions (2CP)**: all traffic flow source nodes are located in two consecutive orbit portions (north-west and north-east portions). It regards the configurations named R_1-R_3-R_5-R_7-R_9-R_{11} for Scenario 1 and Scenario 2, and R_1-R_3-R_5-R_7-R_9-R_{11}-R_{13}-R_{15} for Scenario 3 and Scenario 4.
3) **Two not consecutive portions (2NCP)**: all traffic flow source nodes are located in two not consecutive orbit portions, which are opposite portions (north-west and south-east portions). It regards the configurations named R_1-R_3-R_5-R_{13}-R_{15}-R_{17} for Scenario 1 and Scenario 2, and R_1-R_3-R_5-R_7-R_{17}-R_{19}-R_{21}-R_{23} for Scenario 3 and Scenario 4.
4) **All portions (AP)**: the traffic flow source nodes are equally distributed in all orbit portions. It regards the configurations named R_1-R_3-R_5-R_7-R_9-R_{11}-R_{13}-R_{15}-R_{17}-R_{19}-R_{21}-R_{23} for Scenario 1 and Scenario 2, and R_1-R_3-R_5-R_7-R_9-R_{11}-R_{13}-R_{15}-R_{17}-R_{19}-R_{21}-R_{23}-R_{25}-R_{27}-R_{29}-R_{31} for Scenario 3 and Scenario 4.

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

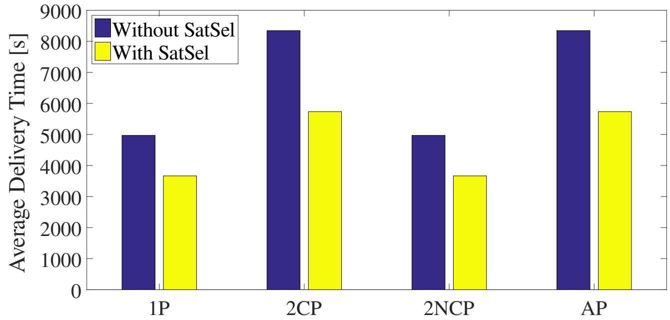
$$ADT = \frac{\sum_{n=1}^{N} \left(T_n^{RX} - T_n^{TX} \right)}{N} \quad (10)$$

where N is the number of bundles per traffic flow, T_n^{RX} is the time instant when the n-th bundle is received by the destination, and T_n^{TX} is the time instant when the n-th bundle is transmitted by the source.
For each simulation, we computed the ADT using three different mechanisms for the nanosatellite selection:

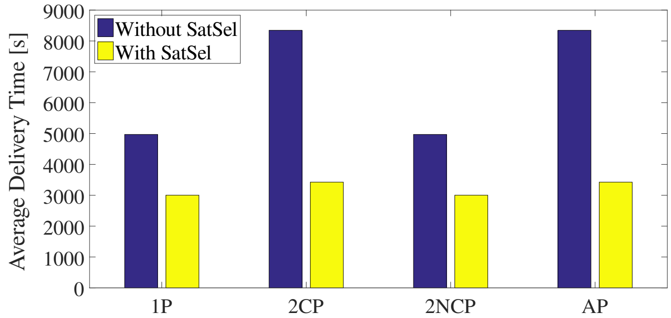
- **next nanosatellite (Without SatSel)**: all bundles are uploaded on the next nanosatellite that will come in contact with the source cold spot, only considering the bound of Q_B bundles for each contact.
- **SatSel with *n* stored value (With SatSel *n*)**: the source cold spot decision is based on nanosatellite buffer occupancy information received during the previous *n* contacts between the source cold spot and each nanosatellite.
- **SatSel with all stored values (With SatSel All)**: the source cold spot decision is based on nanosatellite buffer occupancy information received during all previous contacts between the source cold spot and each nanosatellite.

In order to quantify how the performance changes varying the considered number of previous contacts for the nanosatellite buffer occupancy estimation (nanosatellite buffer occupancy history size), in our simulations the value of *n* changes from 1 to 5.
In the first set of simulations, each traffic flow is composed of a burst of 136 bundles of 50 KB each. We chose this number of bundles because it is the maximum number of bundles that a nanosatellite can upload/download during each contact, also considering a margin time at the begin of each contact necessary to send nanosatellite buffer occupancy information. Figure 3 shows the performance obtained by using this simple traffic flow configuration.

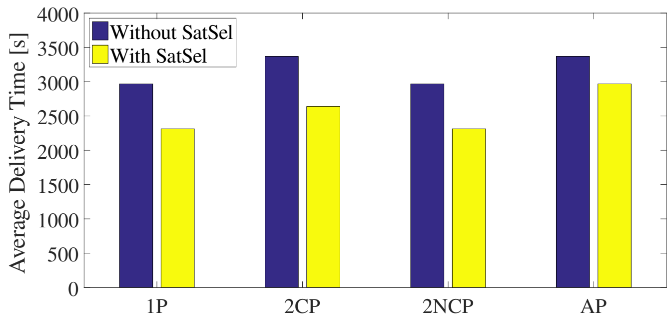
SatSel reduces the ADT in all simulated scenarios and network load configurations. The obtained performance improvement varies depending on the topology and traffic load on the network, and it is ranging between 26% and 32%, 39% and 59%, 12% and 22%, and it is about 34% in Scenario from 1 to 4, respectively.
We also performed a second set of simulations, where the traffic flow configuration is more similar to the one of a real case scenario. In these simulations, each traffic flow is composed by 10 bursts of bundles. Each burst size varies between 100 and 150 bundles, and inter-burst time is the flight time between two consecutive nanosatellites. In this way, we simulate that the amount of bundles each source cold spot has to upload on nanosatellites at the begin of each contact is not constant, in order to quantify the performance improvement introduced by the nanosatellite buffer occupancy history and how it changes varying the nanosatellite buffer occupancy history size.
Figure 4 shows the performance obtained by using this other



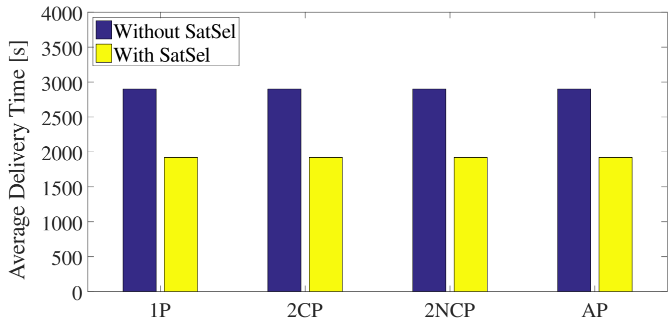
(a) Scenario 1



(b) Scenario 2

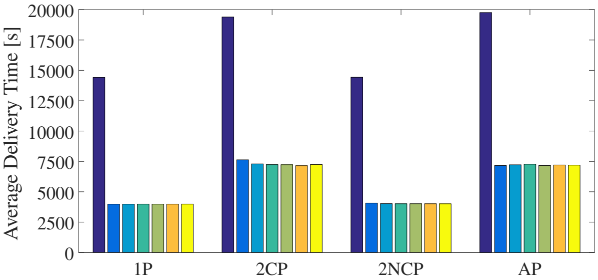


(c) Scenario 3

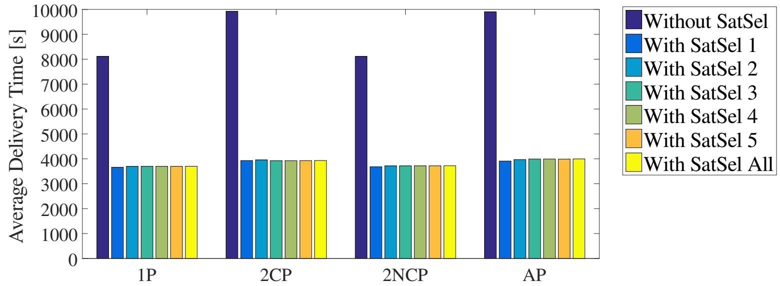


(d) Scenario 4

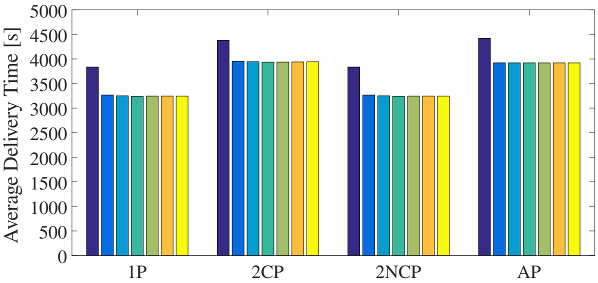
Figure 3: Average delivery times 136 bundles traffic flow configuration



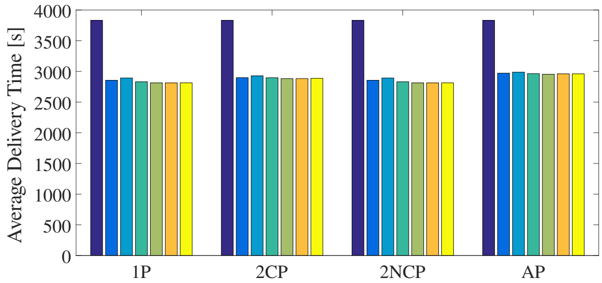
(a) Scenario 1



(b) Scenario 2



(c) Scenario 3



(d) Scenario 4

Figure 4: Average delivery times bursty bundles traffic flow configuration

traffic flow configuration.
Also in this case, SatSel reduces the ADT in all simulated scenarios and network load configurations. The obtained performance improvement is ranging between 60% and 72%, 54% and 60%, 10% and 16%, and 22% and 26% in Scenario from 1 to 4, respectively. However, the obtained performance varying the nanosatellite buffer occupancy history size are quite similar (the difference is less than 3% in all cases). Since the main purpose of this SatSel functionality is to avoid that some possible single anomalous values of the nanosatellite buffer occupancy history strongly affect the nanosatellite buffer occupancy estimation, the reason of this performance trend is that with the used traffic flow configuration all nanosatellite buffer occupancy values do not significantly deviate from the mean value, which is the estimated value.

IV. CONCLUSIONS AND FUTURE WORKS

In this paper we present SatSel, a selection algorithm able to reduce data delivery time in a DTN-nanosatellite rural access network. SatSel allows all cold spots to decide to which nanosatellite upload each bundle destined to the Internet. At the begin of each contact, the cold spots receive information from the nanosatellites in contact with about their buffer occupancy, in order to estimate how many bundles each nanosatellite can upload and store in its buffer and the bundle delivery time for each possible choice, i.e. for each nanosatellite of the network. However, each cold spot cannot know the nanosatellite buffer occupancy if the nanosatellites are not in contact with them. SatSel also estimates these information considering the values of the nanosatellite buffer occupancy when they came in contact with the cold spot during the previous passages. All these values compose the nanosatellite buffer occupancy history.

It is not an optimal algorithm, but it learns from the network over time in order to reduce the percentage of bad choice.

We performed simulations changing the topology of the network (the number and position of nanosatellites and ground stations) and the traffic flow configurations. Moreover, in order to quantify the obtained performance improvement, we calculated the average bundle delivery time with and without SatSel, allowing all rural gateways to calculate how many bundles upload on each nanosatellite or to upload the maximum number of bundles during each contact. We also performed simulations changing the nanosatellite buffer occupancy history size, i.e. the number of previous contacts between each cold spot and each nanosatellite to consider to make the nanosatellite buffer occupancy estimations.

SatSel reduces the average bundle delivery time in all simulated scenarios, obtaining a performance improvement ranges between 10% and 72% depending on the topology of the network and the traffic flow configurations. However, changing the nanosatellite buffer occupancy history size, the obtained performance do not significantly change (the difference is less than 3% in all cases). The reason is that, with the used traffic flow configurations, the nanosatellite buffer occupancy values which composed the nanosatellite buffer occupancy history do not significantly deviate from the estimated value, which is their mean value.

One target of our future work is to verify if the introduction of some thresholds is useful for the bundle delivery time estimation, in order to exclude the previous contacts whose nanosatellite buffer occupancy value are anomalous, which may be caused by unwanted and unexpected events, such as blackouts, quite frequent in underdeveloped countries.

We are also working on the development of a multi-orbit nanosatellites constellation in our simulator, including the study and implementation of an inter-nanosatellite routing, whose purpose is to further reduce the bundle delivery time considering some constraints, such as the available energy and the nanosatellite buffer occupancy during the contacts among them.

REFERENCES

[1] S. Burleigh. Nanosatellites for universal network access. In *Proceedings of the 2013 ACM MobiCom workshop on Lowest cost denominator networking for universal access*, pages 33–34. ACM, 2013.

[2] 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*, 99(11):1980–1997, 2011.

[3] M. Cello, M. Marchese, and F. Patrone. Hot spot selection in rural access nanosatellite networks. In *Proceedings of the 9th ACM MobiCom workshop on Challenged networks*, pages 69–72. ACM, 2014.

[4] M. Cello, M. Marchese, and F. Patrone. Hotsel: A hot spot selection algorithm for internet access in rural areas through nanosatellite networks. In *Proceedings of Global Communications Conference 2015, GLOBECOM '15*. IEEE, 2015.

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

[6] Facebook and partners. Facebook and partner's project internet.org. http://www.internet.org/, 2014.

[7] Google. Google project loon. http://www.google.com/loon/, 2013.

[8] D. L. Johnson, V. Pejovic, E. M. Belding, and G. Van Stam. Traffic characterization and internet usage in rural africa. In *Proceedings of the 20th international conference companion on World wide web*, pages 493–502. ACM, 2011.

[9] OneWeb. Oneweb's project. http://www.oneweb.world/, 2015.

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

[11] SpaceX and partners. Spacex and partner's project. http://www.spacex.com/, 2015.