ColdSel: A Selection Algorithm to mitigate congestion situations over Nanosatellite Networks

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*—Nanosatellites offer a solution to provide Internet access in areas where there is no ICT infrastructure, such as rural and remote areas, keeping low construction and maintenance costs. A nanosatellite constellation is composed of a certain number of nanosatellites distributed in one or more orbital planes. In most cases, this network cannot ensure an end-to-end connection between sources and destinations, because not all satellite links are up at the same time. The Delay and Disruption Tolerant Networking (DTN) paradigm is so necessary to deal with predictable disruptions and large delays, allowing nanosatellites, Internet gateways (also called hot spots), and rural gateways (also called cold spots) to keep data stored in their buffers also for tens of minutes waiting to be forwarded. We focused our attention on a possible selection strategy which indicates to all nanosatellites how many bundles (PDUs of the DTN protocol) they can upload from rural gateways during each contact. The decision is made by considering the maximum amount of data that nanosatellites and cold spots can exchange during each contact (quantity related to contact durations) and the possible presence of congestion situations in rural and remote areas or in nanosatellites. The proposed strategy is called ColdSel: it calculates the amount of bundles that each cold spot has to upload on the nanosatellite in contact with in order to mitigate congestion situations in rural areas and nanosatellites and to reduce the nanosatellite buffer occupancies so improving the performance in terms of data delivery time.**

I. Introduction

Currently, only about 46% of the world population has access to the Internet [1]. The main problem is that a huge amount of people lives in rural areas or countries where there is no ICT infrastructure. There are several different solutions which can be implemented to increase this percentage. In the past few years some industries started projects with this purpose. Google's Project Loon [8] involves the use of a network of balloons travelling at an altitude of about 20 km, Facebook and partners' project Internet.org [7] provides the use of self-powered and self-driving drones, SpaceX and partners' project [11] and the Oneweb project [10] are based on the utilization of a huge number of microsatellites, a type of Low Earth Orbit (LEO) satellites.
Another possible solution involves the use of Nanosatellites [2] 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 and Disruption Tolerant Networking (DTN) paradigm [3], [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). In short, the data forward path is from $R_{i\in\{1,M\}}$ to $CS_{j\in\{1,2\}}$ to $SAT_{k\in\{1,3\}}$ to $HS_{h\in\{1,2\}}$ to the Internet to D. The data reverse path is the opposite.

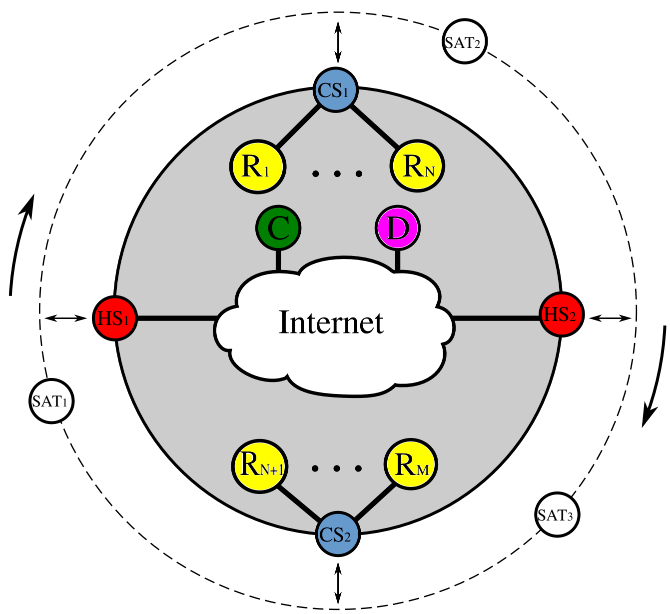


Figure 1: Nanosatellite network scenario.

Since nanosatellites are a type of LEO satellites, nanosatellite footprints are not the same for all orbit time. In order to keep low the overall network cost, the number of nanosatellite could not be enough to guarantee a footprint overlap and so neither to guarantee a satellite link always up between a ground station and a nanosatellite. In this way, there could not be a persistent path between source and destination. The time necessary to get an end-to-end communication is much greater

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

than the one obtained when both endpoints are on the Internet. There are several ways to reduce data delivery time, which is a critical performance parameter for some services. In [4] we developed an hot spot selection algorithm (called HotSel) which reduces the delivery time for all data from the Internet to the rural areas (reverse path). In this article we propose "ColdSel": a selection algorithm which reduces nanosatellite buffer occupancy and delivery time for all data on the path from the rural areas to the Internet (forward path). The purpose of this algorithm is also to reduce the performance decrease due to the possible presence of congestion situations.

II. Motivations

In this kind of network, each contact between a nanosatellite and a ground station has a fixed and known a priori duration. Given a certain transmission bandwidth also the amount of data that can be uploaded and downloaded during each contact is fixed and known a priori.
A congestion situation may happen in two different cases:

1) *cold spot congestion*: the traffic generation rate of one or more rural areas is larger than the nanosatellite constellation upload capacity per rural area, i.e. the total amount of data that all nanosatellites can upload from the cold spot during each orbit.
2) *nanosatellite congestion*: the traffic generation rate of one or more rural areas is larger than the nanosatellite constellation download capacity per rural area, i.e. the total amount of data that all nanosatellites can download to hot spots during each orbit.

For example, let us consider a network composed of 4 HSs, 4 nanosatellites, and 8 CSs, with a maximum number of bundles that each nanosatellite can upload from each cold spot and download to each hot spot set to 100. The nanosatellite constellation download capacity is 1600 bundles, while the nanosatellite constellation upload capacity is 3200 bundles. If we uniformly distribute these two amounts of bundles to all rural areas, each cold spot will be allowed to upload up to 200 bundles (50 bundles during each contact) during each orbit to avoid a constant growth of the nanosatellite buffer occupancy, and up to 400 bundles (100 bundles during each contact) to avoid congestion of the cold spot buffer occupancy. In this case, if a rural area bundle generation rate becomes larger than 200 bundles per orbit but lower than 400, there will be a cold spot congestion, while if it becomes larger than 400, there will be both cold spot and nanosatellite congestions.
ColdSel dynamically distributes the nanosatellite constellation download and upload capacity to all rural areas depending on the cold spot and nanosatellite buffer occupancies and on the contact volume, computing the number of bundles that each nanosatellite has to upload during each contact with each rural gateway. In this view, ColdSel will run on nanosatellites.
Being directly proportional to rural areas bundle generation rates, the role of the cold spot buffer occupancy should be quite clear. Some more words may be helpful concerning nanosatellite buffer occupancy and related cold spot actions. In the network described in [2], the satellite transceiver acts in the S-Band and can transmit at 230 kbps, so the satellites can upload up to 7.4 MB during each contact, being the contact time about 256 s. However, the available transmission rate using a Ka-band, Q-band, or V-band transceiver is of 30-40 Mbps [9] and even up to 100 Mbps by 2020 [6]. Using these transceivers, if the total rural areas bundle generation rate exceeds the nanosatellite constellation download capacity even for a short time, the nanosatellite buffer occupancies could increase very quickly, with a consequently loss of bundles due to buffer overflow. ColdSel forces bundles to wait most part of the delivery time in cold spot buffers instead of nanosatellite buffers, obtaining lower nanosatellite buffer occupancies but higher cold spot buffer ones. This is preferable because the buffer size is a more critical parameter for nanosatellites than for ground stations.
As far as the contact volume, i.e. the maximum amount of data each nanosatellite can upload/download during each contact, is concerned: it depends on the bandwidth of the satellite links and the contact durations. The contacts between nanosatellites and hot spots are the only opportunities the nanosatellites have to download the bundles stored in their buffers and destined to the Internet. So it is a topical parameter for performance. If a nanosatellite comes in contact with a hot spot and its buffer occupancy is less than the contact volume, it could be possible that a not ideal choice has been made. For example, when the nanosatellite came in contact with one or more rural areas it would not upload enough bundles even though it was possible. So ColdSel must compute the number of bundles that the nanosatellite has to upload from the cold spot low enough to mitigate unbalanced cold spot buffer occupancies, but also high enough to avoid waste of bandwidth in contacts between nanosatellites and hot spots.
In short, the purpose of ColdSel is:

- to mitigate congestion situations on the cold spots, in order to avoid waste of time waiting in cold spot buffers and loss of bundles due to cold spot buffer overflow;
- to reduce nanosatellite buffer occupancy, in order to avoid loss of bundles due to nanosatellite buffer overflow;
- to efficiently use contact volumes, in order to not waste bandwidth in the contact between nanosatellites and hot spots.

The practical effect will be to improve the performance in terms of data delivery time.

III. Data amount calculation

At the begin of each contact between a nanosatellite SAT_K and a cold spot CS_J, the first quantity ColdSel calculates is:

$$DC^J = \left\lceil \frac{DS^{J,INT}}{\sum\limits_{j=1}^{nCS} DS^{j,INT}} * \sum_{i=1}^{nHS} Q^{i,K} \right\rceil, \tag{1}$$

where $DS^{J,INT}$ is the amount of data stored in the CS_J buffer and destined to the Internet, $Q^{i,K}$ is the maximum amount of data that SAT_K can download to HS_i during each contact between them (contact volume), and nCS and nHS are the number of cold spots and hot spots in the network, respectively. DC^J is an indication about the amount of data CS_J has to

upload on SAT_K in order to avoid or, at least, to mitigate congestion situations. In this way, if a cold spot buffer occupancy is larger than another one ($DS^{J,INT} > DS^{j,INT}$), also the value of its DC is greater ($DC^J > DC^j$).
However, the value DC^J cannot be greater than $Q^{J,K}$, i.e. the maximum amount of data that CS_J can upload on SAT_K during each contact between them. If it is so, there is a data capacity surplus ($DC^J - Q^{J,K}$) which can be distributed to all other cold spots. To do this, at the begin of the contact between CS_J and SAT_K, ColdSel computes not only DC^J but all $DC^j, \forall j \in \mathcal{CS}$, where $\mathcal{CS}$ is the set of all cold spots. Afterwards, it updates these values considering the data capacity surplus following the procedure described by the function *ComputeDCValues* in Figure 2.

foreach cold spot $j \in \mathcal{CS}$ **do**
    $DC^j \leftarrow \left\lceil \frac{DS^{j,INT}}{\sum_{j=1}^{nCS} DS^{j,INT}} * \sum_{i=1}^{nHS} Q^{i,K} \right\rceil$;
do
    $D_{SURPLUS} \leftarrow 0$;
    $n_{CSUT} \leftarrow 0$;
    foreach cold spot $j \in \mathcal{CS}$ **do**
        if $DC^j > Q^{j,K}$ **then**
            $D_{SURPLUS} \leftarrow D_{SURPLUS} + (DC^j - Q^{j,K})$;
            $DC^j \leftarrow Q^{j,K}$;
        else
            $n_{CSUT} \leftarrow n_{CSUT} + 1$;
    if $D_{SURPLUS} > 0$ **then**
        foreach cold spot $j \in \mathcal{CS}$ **do**
            if $DC^j < Q^{j,K}$ **then**
                $DC^j \leftarrow DC^j + \frac{D_{SURPLUS}}{n_{CSUT}}$;
while $D_{SURPLUS} > 0$;

Figure 2: Function *ComputeDCValues*

To collect all the information about cold spot buffer occupancies ($DS^{j,INT}, \forall j \in \mathcal{CS}$), we defined a protocol which allows cold spots to send these information to the nanosatellites at the begin of each contact among them. However, each nanosatellite can know the buffer occupancy of each cold spot only when it enters in contact with them. In this way, each nanosatellite has to wait an entire orbit time to know the buffer occupancy of all cold spots, and, after one entire orbit time, some information can be outdated. To attenuate this problem, we also defined a second protocol which allows nanosatellites to send all collected information to the central node C through the hot spots. The central node will merge all these information and will forward back to each nanosatellite a bundle containing all cold spot buffer occupancy values collected by all nanosatellites. In this way, the ColdSel learning phase is reduced, and each nanosatellite will update its information about each cold spot buffer more often than once per orbit.
The second quantity that ColdSel must compute is:

$$DW^J = \left\lceil \frac{\sum_{i=1}^{nHS} Q^{i,K} - DS^{K,INT}}{nCS} \right\rceil, \qquad (2)$$

DW^J is an indication about the amount of data that CS_J has to upload on SAT_K in order to avoid waste of bandwidth in the contacts between nanosatellites and hot spots. When a nanosatellite comes in contact with a hot spot HS_I, the amount of data stored in its buffer and destined to the Internet should be at least $Q^{I,K}$, because if it is smaller, the nanosatellite may perhaps have uploaded more bundles from the previous cold spots (see previous section).
The amount of data to be uploaded, D^J, that SAT_K communicates to CS_J is the maximum between DC^J and DW^J:

$$D^J = \max(DC^J, DW^J), \qquad (3)$$

IV. Performance Analysis

We implemented ColdSel in our DTN module within Network Simulator 3 (NS3), whose characteristics are brefly described in [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}).

In Figure 3 is shown the most complex of the simulated scenarios (Scenario 4).
In our tests, 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. All traffic flows have a fixed number of bundles with the same size M, and all contacts have the same duration and, consequently, the same contact volume. Considering these assumptions, equations (1), (2), and (3) can be rewritten in order to calculate the amount of bundles to upload as:

$$BC^J = \left\lceil \frac{BS^{J,INT}}{\sum_{j=1}^{nCS} BS^{j,INT}} * P * nHS \right\rceil, \qquad (4)$$

$$BW^J = \left\lceil \frac{P * nHS - BS^{K,INT}}{nCS} \right\rceil, \qquad (5)$$

$$B^J = \max(BC^J, BW^J), \qquad (6)$$

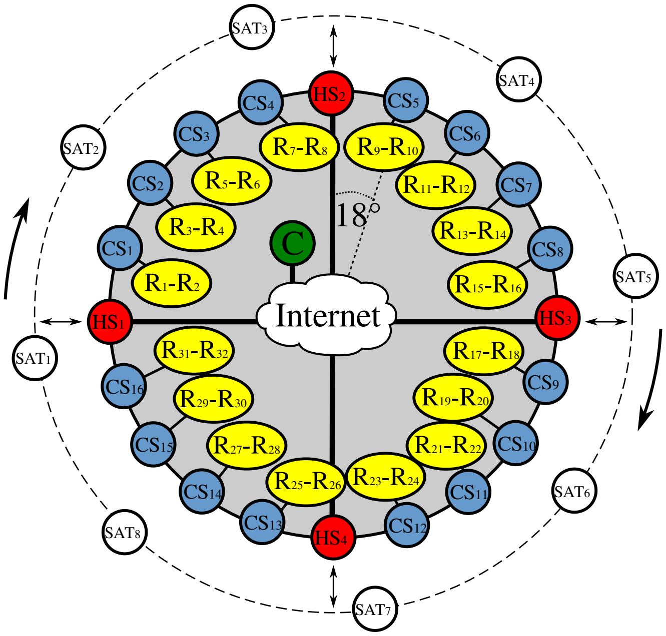


Figure 3: Scenario 4

where P is the number of bundles that can be exchanged between nanosatellites and ground stations during each contact. Also the function *ComputeDCValues* in Figure 2 can be rewritten in the same way.

The satellite link parameters are as in [2]. The nanosatellites altitude is set to 200 km, consequently the orbit time is about 90 minutes and the contact time is about 256 s. A distance threshold is set to calculate the contact time, which corresponds to a minimum value of the elevation angle: the contact begins when the distance between a ground station and a nanosatellite is below this threshold, and ends when their distance returns above it. The S-band transceiver transmission rate is 230 kbps, so the contact volume is about 7 MB. This amount is an underestimation, because we set a margin time for sending the cold spot buffer occupancy information. During contact time, about 140 bundles of 50 KB each can be forwarded between a couple nanosatellite/ground station.

For each scenario, we performed different simulations changing the network congestion configuration, i.e. the number and position of the source nodes and the amount of bundles of each traffic flow, in order to test the performance introducing congestion situations in different rural areas. Since the main practical purpose of ColdSel is to reduce the delivery time of bundles from rural areas to the Internet, all source nodes are always located in rural areas, i.e. are rural nodes. The destination node is always the node C. We made this choice because the delivery time of the path between node C and D is negligible compared to the delivery time from the source rural node to C.

We define as orbit portion each of the four portions obtained dividing the orbit with a vertical line between the northernmost and the southernmost points, and an horizontal line between the westernmost and the easternmost points. These four portions are called north-east, north-west, south-east, and south-west portions. For example, looking at Figure **??**, the north-west portion includes the cold spots from CS_1 to CS_4, the north-east portion the cold spots from CS_5 to CS_8, the south-east portion the cold spots from CS_9 to CS_{12}, and the south-west one the cold spots from CS_{13} to CS_{16}.

In all simulations, each rural area generates a burst of 1000 bundles of 50 KB each. For each network congestion configuration, the rural areas located in determined orbit portions generate twice the traffic of the other areas (i.e. a burst of 2000 bundles). The following formalism has been used: the notation CS_x-CS_y-CS_z-CS_t means that the congested rural gateways are CS_x, CS_y, CS_z, and CS_t at the same time.

To better quantify the performance improvement achievable by using ColdSel, we decided to test four different network congestion configurations:

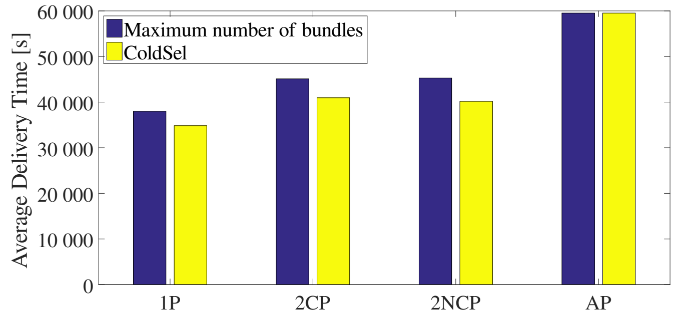
1) **One portion (1P)**: all congested rural areas are located into the same orbit portion (the north-west one). 1P regards the configurations named CS_1-CS_2-CS_3 for Scenarios 1 and 2, and CS_1-CS_2-CS_3-CS_4 for Scenarios 3 and 4.
2) **Two consecutive portions (2CP)**: all congested rural areas are located into two consecutive orbit portions (the north-west and north-east ones). 2CP regards the configurations named CS_1-CS_2-CS_3-CS_4-CS_5-CS_6 for Scenarios 1 and 2, and CS_1-CS_2-CS_3-CS_4-CS_5-CS_6-CS_7-CS_8 for Scenarios 3 and 4.
3) **Two not consecutive portions (2NCP)**: all congested rural areas are located into two not consecutive orbit portions (the opposite portions north-west and south-east). 2NCP regards the configurations named CS_1-CS_2-CS_3-CS_7-CS_8-CS_9 for Scenarios 1 and 2, and CS_1-CS_2-CS_3-CS_4-CS_9-CS_{10}-CS_{11}-CS_{12} for Scenarios 3 and 4.
4) **All portions (AP)**: all rural areas have the same congestion level. It regards the configurations named CS_1-CS_2-CS_3-CS_4-CS_5-CS_6-CS_7-CS_8-CS_9-CS_{10}-CS_{11}-CS_{12} for Scenarios 1 and 2, and CS_1-CS_2-CS_3-CS_4-CS_5-CS_6-CS_7-CS_8-CS_9-CS_{10}-CS_{11}-CS_{12}-CS_{13}-CS_{14}-CS_{15}-CS_{16} for Scenarios 3 and 4.

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

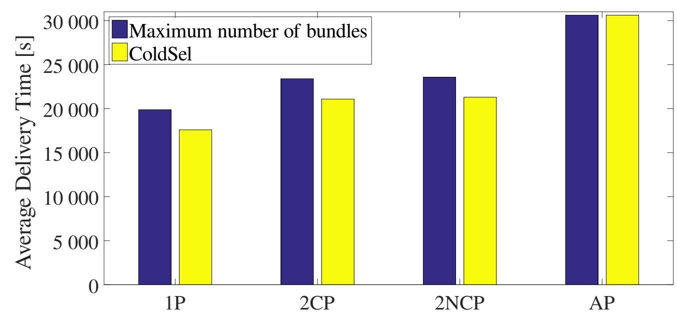
$$ADT = \frac{\sum_{n=1}^{N} \left(T_n^{RX} - T_n^{TX}\right)}{N} \tag{7}$$

where N is the number of bundles per 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.

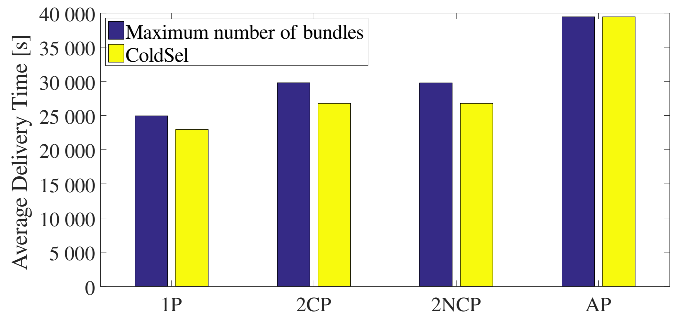
In order to verify and quantify the obtained performance improvement, we computed the ADT using and not using ColdSel for each test, allowing, in this last case, the cold spots to send the maximum number of bundles during each contact. Figure 4 shows the obtained performance. ColdSel reduces the ADT in all simulated scenarios in three out of the four network congestion configurations (1P, 2CP, and 2NCP). The obtained performance improvement is almost similar in all simulated scenarios, and it is ranging between 7% and 12%. The ADT does not change by using or not using ColdSel for the AP configuration simulations, since all rural areas generate the same number of bundles and the cold spot buffer occupancies are always the same. The ColdSel scheme, which explicitly uses the status of the buffers to made decisions, does not



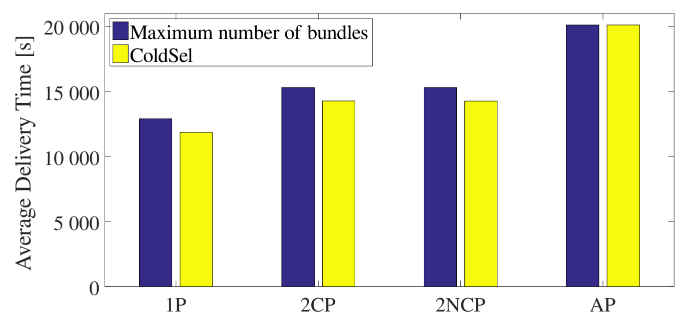
(a) Scenario 1



(b) Scenario 2

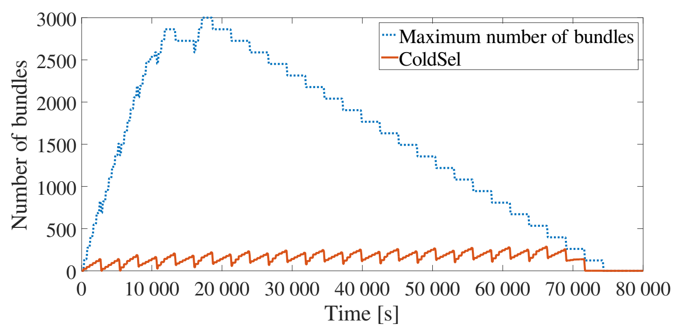


(c) Scenario 3

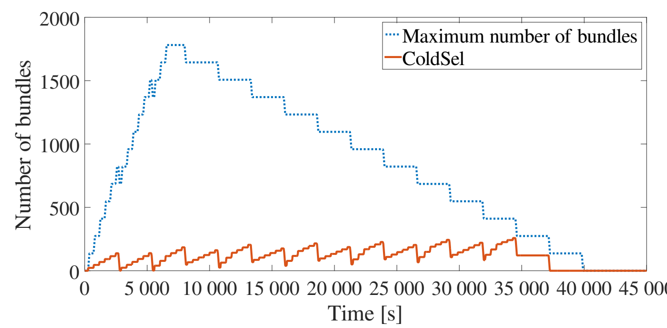


(d) Scenario 4

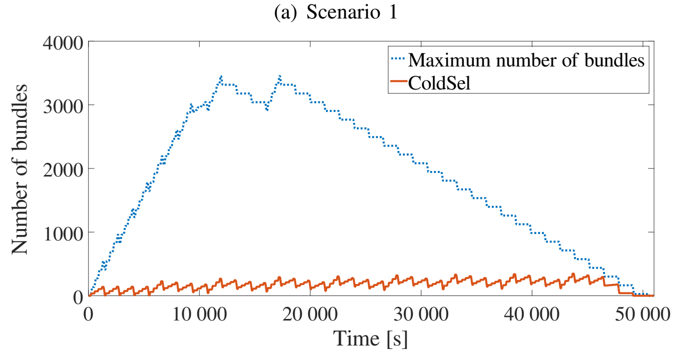
Figure 4: Average delivery times varying network congestion configuration



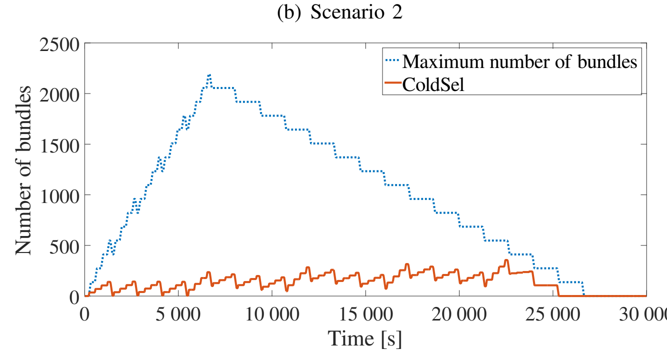
(a) Scenario 1



(b) Scenario 2



(c) Scenario 3



(d) Scenario 4

Figure 5: SAT_1 buffer occupancy trend varying network congestion configuration

provide any advantage in this case. The second parameter we analysed is the nanosatellite buffer occupancy. As previously mentioned, the S-band transceiver transmission rate is 230 kbps, but it could increase up of 30-40 Mbps by using a Ka-band, Q-band, or V-band transceiver. Consequently, also the contact volume would increase up to 0.9-1.2 GB, which means about 140 bundles of 6.7-9 MB each.

Figure 5 shows the nanosatellite buffer occupancy trend of one nanosatellite (SAT_1) in all performed simulations. We decided to not show the other nanosatellite buffer occupancies since all have the same shape and do not give additional information. The SAT_1 maximum buffer occupancy value using ColdSel is much lower than without ColdSel. This value using ColdSel is 286, 258, 348, and 356, while without ColdSel is 3000, 1781, 3452, and 2192 in Scenario from 1 to 4, respectively. The obtained reduction is ranging between 84% and 90%.

Another important aspect worth noting in Figure 5 is the trend. Allowing the cold spots to send the maximum number of bundles for each contact implies a big variation on time of the nanosatellite buffer occupancy, while ColdSel keeps the occupancy almost constant. In this work, we did not consider buffer size constraints: we assumed infinite buffers. However, ColdSel may help avoid congestion situations and optimize nanosatellite buffer occupancies also in case of finite buffer size.

V. Conclusions and Future Works

In this paper we presented ColdSel, an algorithm able to mitigate possible congestion situations, to limit the nanosatellite buffer occupancy and to reduce the data delivery time in a DTN-nanosatellite rural access network. At the beginning of each contact between nanosatellites and cold spots, ColdSel allows each nanosatellite to compute the maximum amount of data that the cold spot which it is in contact with can send. We also defined some ad-hoc protocols whose purpose is to allow nanosatellites to collect all the required dynamic information, which are the cold spot buffer occupancies and the contact volumes. These information are then forwarded by each nanosatellite through the hot spots to the central node which merges them all and sends the result back to each nanosatellite. In this way, all nanosatellites can know the buffer occupancies of all cold spots before entering in contact with each of them.

We performed simulations changing the topology of the network (the number and position of nanosatellites and ground stations) and the network congestion configuration (the number and position of congested rural areas). Moreover, in order to quantify the obtained performance improvement and the reduction of nanosatellite buffer occupancies, we showed the average delivery time and the nanosatellite buffer occupancy trend with and without ColdSel.

ColdSel reduces the average delivery time in all simulated scenarios, obtaining a performance improvement ranging between 7% and 12% in three out of the four network congestion configurations. It also reduces the average nanosatellite buffer occupancies, avoiding possible data bundles losses due to nanosatellite buffer overflows. The obtained reduction of the maximum buffer occupancy value is ranging between 84% and 90%.

ColdSel has a drawback related to the reliability of the cold spot buffer occupancy information. The time interval between the instant when a nanosatellite obtains the buffer occupancy information of a cold spot and the instant when this information is used by ColdSel could be of the order of magnitude of minutes, and, consequently, could be outdated, especially if the rural area bundle generation rate is high or extremely variable.

A larger set of simulations will be performed in future in order to better estimate which is the relation between the obtained performance improvement and the delay regarding the use of the collected cold spot buffer occupancy information changing the bundle generation rate.

We are also working on the development of a multi-orbit nanosatellite constellation, 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 occupancies during the contacts among them.

References

[1] Internet usage statistics. http://www.internetworldstats.com/stats.htm, Nov. 2015.

[2] 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.

[3] 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.

[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] E. Commission. Europe 2020 flagship initiative, innovation union, sec(2010) 1161. http://ec.europa.eu/research/innovation-union/pdf/innovation-union-communication_en.pdf, 2010.

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

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

[9] A. Kyrgiazos, B. Evans, P. Thompson, P. T. Mathiopoulos, and S. Papaharalabos. A terabit/second satellite system for european broadband access: a feasibility study. *International Journal of Satellite Communications and Networking*, 32(2):63–92, 2014.

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

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