Enabling Green Networking with a Power Down Approach in LEO Satellite Constellations

Conference Paper · June 2017

DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.78

6 authors, including:

Mohammed Hussein
Kirkuk University
5 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Iyad Tumar
Birzeit University
29 PUBLICATIONS **259** CITATIONS
SEE PROFILE

Wasel Ghanem
Birzeit University
16 PUBLICATIONS **122** CITATIONS
SEE PROFILE

Sobhi Ahmed
Paris8 University
7 PUBLICATIONS **217** CITATIONS
SEE PROFILE

2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom)
and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

Enabling Green Networking with a Power Down Approach in LEO Satellite Constellations

Mohammed Hussein, Israa Elayyan, Iyad Tumar, Abdellatif Abu-Issa, Wasel Ghanem, Sobhi Ahmed
Birzeit University, Palestine
mhussein@birzeit.edu, israaelayyan95@gmail.com, itumar@birzeit.edu, abuissa@birzeit.edu, ghanem@birzeit.edu,
sahmed@birzeit.edu

***Abstract*—According to several studies, for batteries of the type that powered by Iridium NEXT satellites, a 15% reduction in depth of discharge (DoD) the battery lifetime almost doubles. Given the satellites in LEO constellations can spend 30% of their time under the earth‘s umbra, time during which they are powered by batteries. Also, due to several geographical and economic flourish constraints, some Inter-Satellite links (ISLs) are expected to be loaded with data packets while others remain unused.**

Most research on satellite power management are at component level or link level, treating satellites as isolated devices. A complementary approach is to facilitate power management at constellation level by routing traffic through different paths to adjust the workload on individual links. Since these networks are typically over-provisioned, selectively shutting down satellite links during periods of low demand seems like a good way to reduce energy consumption. While still guaranteeing full connectivity and maximum link utilization constraints. Since identifying the optimal set of links to shut down is an NP-complete problem that can be solved only for trivial cases, therefore, we propose efficient heuristics that can be used for LEO satellite network topology.

I. INTRODUCTION

Recently significant research efforts have been focused on reducing the energy consumption of Information and Communication Technology (ICT) networks. There is a significant literature body on power-awareness in wireless and mobile networks [1], [2] and wired network [3]. However, many challenges need to be addressed to develop and deploy energy efficient LEO-satellite network.

In this paper, we aim at studying how to reduce the overall power consumption of a LEO satellite constellation, considering it as a single, large and distributed system. We do not focus on reducing the power consumption of each satellite [4], which has been introduced over years, but rather we aim at controlling the whole constellation, so as to find the minimum set of satellite nodes that must be used to meet the actual traffic matrix demand. A network management system that exploits opportunities to selectively power on network satellites, based on the current or projected traffic matrix. The network management system must solve an optimization problem that takes the LEO network topology and traffic matrix as input, and identifies the minimum number of network nodes that can be powered on while still guaranteeing full connectivity and Quality of Service (QoS) constraints. Since finding the optimal solution is an NP-complete problem, we propose two heuristic algorithms whose performance we evaluate in a realistic traffic matrix. To the best of our knowledge, this is the first work that considers the extending satellite service life by turning off satellite nodes in a LEO satellite constellation. The work that is closest in spirit to this work is our previous work [5] on routing for extending satellite service life in LEO satellite networks. The similarities between the two approaches that, both approaches give a penalty to use the satellites located in earth’s umbra.
Simulation results based on LEO satellite constellation and real traffic matrices show that, MNH and MENH heuristics can increase the percentage of power saving 23% and 51%, respectively. This will lead to decrease the depth of discharge in eclipsed satellites batteries, then, increasing the LEO satellite constellation availability as well. As the data shows, this improvement is accomplished by trading off very little in terms of end-to-end delay.

The rest of the paper is organized as follows. Section II overviews the related work. In section III, we present the traffic modeling and problem formulation and in section IV we describe our heuristics. Section V details the results obtained. Finally, conclusions are drawn in section VI.

II. RELATED WORK

Traditionally, energy efficiency has not been a major consideration in networking design. However, as reducing energy consumption has emerged as a major scientific and engineering challenge the subject has attracted more attention from the ICT industry. In a pioneering work [1], Gupta and Singh discussed the benefits of powering down network components and the impact this could have on the network protocols. In a closely related work [2], the authors considered different approaches for switching off a specific number of network elements (nodes and links) in backbone networks while still ensuring end-to-end connectivity. Unfortunately, these approaches cannot be readily applied to our context as they ignore the specifics of the LEO satellite constellations. In [4], the energy allocation and admission control problem of a single satellite is discussed. The authors use dynamic programming to minimize the energy cost, subject to various delay constraints such as the deadline by which all packets must be transmitted. Satellite NDMA (S-NDMA) [6] uses the channel statistical information to compute the number of packet transmissions separated by a round trip time (RTTs) such that the energy consumption is minimized and a set of QoS requirements is met. However, these schemes consider the energy allocation of a single satellite not the satellite constellation as a whole. In [5] the authors considered the

978-1-5386-3066-2/17 $31.00 © 2017 IEEE
DOI 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.78

problem of extending the satellite battery lifetime by reducing the energy consumption of eclipsed satellites. It was shown that routing traffic away from eclipsed satellites reduces their batteries depth of discharge, thereby significantly increasing their lifetimes. However, the possibility of selectively shutting down the underutilized transceivers of eclipsed satellites was not considered. Note that, an idle transceiver can consume almost half the amount of energy it takes to transmit and receive data traffic.

To the best of our knowledge, this is the first work to consider the problem of network pruning in a satellite constellation for extending its lifetime. Our work is motivated by the observation that LEO satellite constellations are over-provisioned in terms of capacity as individual satellites face highly uneven traffic demands. The population density, and by extension the customer base, is high in cities, low in rural areas and almost zero over the oceans (around 70% of the earth's surface) [7].

III. TRAFFIC MODELING AND PROBLEM FORMULATION

A. Traffic Model

In this section, we describe how we determine the real traffic matrix in a LEO satellite constellation [8], [9].

Using the Virtual Node concept, the whole world is divided into 6×12 cells – each cell occupying 30° latitude and 30° longitude. The traffic matrix depends on the statistics about the user density level per cell, Internet host density levels per continent, and user activity levels per hour. The traffic demand between satellites s and d, i.e, t^{sd}, depends on the user traffic density level, u_s, the host density level, h_d, and the distance, $l(s,d)$, between the satellites:

$$t^{sd} = \frac{(u_s \cdot h_d)^{\alpha}}{(l(s,d))^{\beta}} \tag{1}$$

Where s corresponds to the LEO logical location (n,m), with $n = \lceil \frac{s}{M_L} \rceil$, $m = s\ MOD\ M_L$ and M_L being the number of satellites in a LEO plane. The distance between two zones can be calculated using the longitude and latitude for the centers of those zones. Finally, we use the values $\alpha = 0.5$ and $\beta = 1.5$, recommended in [9]. To adapt the generated traffic model to practical wideband LEO satellite network, we use t^{sd} as the proportional coefficient for obtaining the average traffic values T^{sd} between two satellites.

$$T^{sd} = \frac{t^{sd}}{\sum\limits_{\forall s}\sum\limits_{\forall d} t^{sd}} \times \frac{\text{total offered traffic}}{3600} \times \frac{a_h}{100} \tag{2}$$

Here the total offered traffic represents the total traffic generated worldwide per day, with a_h representing the activity percentage during hour h. Note that the average traffic demand is not only a function of the location of the source-destination pair but it is also a function of the time slot.

We have implemented the traffic generation algorithm in Matlab and Fig. 1 shows the results for the arrival rates for different traffic zones at a given time when the total offered traffic is 500Tb/day [10]. As the results show, satellites covering the Northern Hemisphere have larger arrival rates than those covering the Southern Hemisphere.

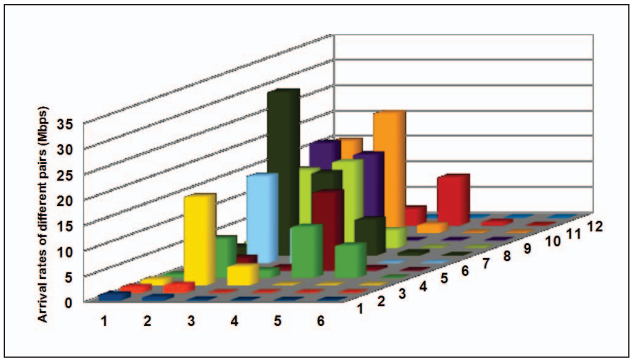

Figure 1: Distribution of traffic arrival rates at a given time for each of the 72 satellites comprising the LEO constellation.

B. Problem Definition

We model the LEO network as un undirected weighted graph $G(V,E)$, where V is the set of $N = |V|$ nodes and E is the set of $L = |E|$ links. Each link $(u,v) \in E$ between two nodes $u, v \in V$ has a capacity c_{uv}. Satellite nodes can be divided into two categories based on whether they are exposed to the sun or eclipsed by the earth. Our interest is obviously on the eclipsed satellites powered by batteries.

Let p_{uv} be the power consumption of the *eclipsed* link (u,v) and x_{uv} be a binary variable denoting whether link (u,v) is pruned or not. The objective is to find a network pruning that minimizes the total power consumption of the eclipsed links in the constellation. The problem can be formulated as follows:

$$minimize\ P_{total} = \sum_{uv=1}^{size(\mathbf{L}_{eclipse})} p_{uv} x_{uv} \tag{3}$$

Such that:

$$\sum_{v=1}^{N} f_{uv}^{sd} - \sum_{v=1}^{N} f_{vu}^{sd} = \begin{cases} T^{sd} & \forall s,d, \quad u = s \\ -T^{sd} & \forall s,d, \quad u = d \\ 0 & \forall s,d, \quad u \neq s,d \end{cases} \tag{4}$$

$$\sum_{s=1}^{N}\sum_{d=1}^{N} (f_{uv}^{sd} + f_{vu}^{sd}) \leq \alpha c_{uv} x_{uv} \qquad \forall u,v \tag{5}$$

Where $\mathbf{L}_{eclipse}$ is the set of the eclipsed links, $\alpha \in (0,1)$ the maximum link utilization that can be tolerated, T^{sd} the average amount of traffic going from satellite node s to satellite node d, f_{uv}^{sd} the amount of flow from s to d that is routed through the arc from u to v and, finally, f_{uv} the total amount of traffic flowing on the link from u to v.

Equation 3 quantifies the total power consumed by all active eclipse links. Equation 4 is the standard flow conservation constraint that ensures no flow is lost and ensures that the sum of the flows leaving the source satellite or entering the destination satellite sums to be T^{sd}. Equation 5 forces the link load to be smaller than the maximum link utilization ratio, α.

This problem formulation falls in the class of capacitated multi-commodity minimum cost flow problems (CMCF), well-known to be NP-hard [11]. Therefore, we propose heuristics for computing approximate solutions in polynomial time.

IV. EFFICIENT HEURISTICS

Given the NP-complete formulations presented, finding the optimal solution using an ILP solver need a lot of time. Therefore, heuristic approaches have be adopted to solve the problem in a reasonable time. In this section, we present two heuristics explicitly designed to find an admissible solution. The heuristics add satellite nodes in certain order until all the input demands can be served. The two heuristics differ in the order in which they consider the satellite nodes. We use a simple shortest path algorithm in our heuristics.

In the begining, the two heuristics start by considering a network in which all satellite nodes are powered off, hence $y_u = 0 \, \forall u$. Then, the algorithm check iteratively to turn on satellite nodes until all the demands list will be served.

A. *Minimum Nodes heuristic (MNH)*

The first heuristic is called Minimum Nodes Heuristics (MNH). The aim of MNH is to turn on the minimum number of satellite nodes so that all flows are served. Unlike other networks such as sensor networks where battery lifetime is also essential, the movement of the satellites in a LEO constellation, such as Iridium, is deterministic. The location of any satellite can be computed and so can if a satellite is eclipsed [12]. MNH algorithm does not leverage this characteristic to promote the choice of turning on satellites that located in eclipse than satellites that located in sun. The metacode for MNH is proposed in algorithm. 1. Therefore, the next heuristic improves on MNH by making more careful choices.

Let *S*= $<Demand_{List}$,$Nodes_{List}>$ describes the current state of the nodes that are needed to be turned on in order to satisfy the current demands in $Demand_{List}$ with their demand's constraints. The possible elements of the $Demand_{List}$ are [D_1,D_2,..., D_m], where $D's$ symbol donates to a demand, and m is the number of demands. In similar way, the $Nodes_{List}$ are [N_1,N_2,..., N_k], where $N's$ donates to a node in the network that should be turned on to serve the demands in the $Demand_{List}$, and k is the number of network nodes.

Let $Input\_Demands$ is a list of demands are required to be served by a particular set of nodes, where such set is a subset from the network nodes set $Network\_Nodes$. Assume that $\forall d \in Input\_Demands$ has at least one shortest path SP_{ij} can serve d, where SP_{ij} represents the jth minimal set of nodes, shortest path, that can meet the ith demand's specification in the $Input\_Demands$ set. Note that it is possible to have more than one shortest path for the same demand.

To complete the formalization of the problem, it is mandatory to define the initial state, goal state, the operator with its precondition and postcondition, and the cost function. The initial state $x_{initial} =< \{\}, \{\} >$ has empty list of the served demands, and empty list of nodes as well. On the other side, the goal state $x_{goal} =< Input\_Demands, \_ >$ should serve all input demands by turning on a particular set of nodes determined by searching in the state space. $add(SP_{ij})$ is an operator which adds the jth shortest path of nodes of the ith demand on a state x. The precondition and postcondition of the add operator are given as follows:

$$Operator : add(SP_{ij})$$

$$Currentstate : x =< Demand_{List}, Nodes_{List} >$$

$$Precondition : DemandD_i \notin Demand_{List}$$

$$Postcondition : Demand'_{List} = Demand_{List} \cup D_i,$$

$$Nodes'_{List} = SP_{ij} \cup Nodes_{List}$$

$$NewState : x' =< Demand'_{List}, Nodes'_{List} >$$

The cost function of applying the add operator on the state x is defined by the number of the *new nodes* that have been added on the state x. In more formal way, the cost can be written as:

$$Cost = |Nodes'_{List}| - |Nodes_{List}|$$

Algorithm 1 Minimum Nodes Heuristic (MNH)

```
Require: $Initial\_state(x_{initial})$, $Goal\_state(x_{goal})$,
    $Input\_Demands\_List(D_{in})$,
    $Max\_Number\_Of\_Shortest\_Paths\_Per\_Demand(K_{max})$
if $x_{initial} == x_{goal}$ then
    $return\ x_{goal}.Nodes\_list$
end if
$Demands\_shortest\_paths\_dictionary(DSPD) = \{\ ,\}$
for i=1 to size($D_{in}$) do
    if $K_{max} <= 0$ then
        $SP\_Of\_Demand = get\_all\_shortest\_paths\_(D_i)$
    else
        $SP\_Of\_Demand = get\_first\_K_{max}\_of\_shortest\_paths\_(D_i)$
    end if
    $DSPD.add(D_i, SP\_Of\_Demand)$
end for
$Initialize\ Queue(Q)\ by\ (x_{initial}, cost = 0)$
while $true$ do
    $x = dequeue\_lowest\_cost\_state(Q)$
    if $size(Q) == 0$ then
        $Return\ No\ Solution$
    end if
    if $x == x_{goal}$ then
        $return\ x.Nodes\_list$
    end if
    for i=1 to size($DSPD.Keys$) do
        $Demand = DSPD.Keys(i)$
        if $Demand \in x.Demand_{List}$ then
            $Continue$
        end if
        $List\_SP = DSPD.Value(Demand)$
        for j=1 to size($List\_SP$) do
            $x_{new} = add(List\_SP(j), x)$
            $Old\_Len = length(x.Nodes\_list)$
            $New\_Len = length(x_{new}.Nodes\_list)$
            $cost = New\_Len - Old\_Len + x.cost$
            $Q.enqueue(x_{new}, cost)$
        end for
    end for
end while
```

B. Minimum Eclipsed-Nodes Heuristic (MENH)

To improve over the MNH, the MENH algorithm tries to switch on satellite nodes that located in sun (powered by solar panel), than satellite nodes that located in earth‘s umbra. Since, satellites in LEO constellations like Iridium can be under the earth‘s eclipse around 30% of the time, making batteries essential to their operation. While the batteries are recharged by solar energy, their lifetime is highly affected by the depth of discharge [13], [14]. The MENH heuristic is similar to MNH in that both algorithms try to turn on the minimum number of satellite nodes that should serve all input demands. However, MENH heuristic tries firstly to turn on satellite nodes that located in sun, decrease the DoD in eclipse satellites battery. The metacode for MENH proposed in algorithm 2. The cost function for MENH is as follows:

$$Cost = |Nodes'_{List} \cap Eclipse\_Nodes\_Set| - \\ |Nodes_{List} \cap Eclipse\_Nodes\_Set|$$

Algorithm 2 Minimum Eclipsed-Nodes Heuristic (MENH)

```
Require: $Initial\_state(x_{initial})$, $Goal\_state(x_{goal})$,
    $Input\_Demands\_List(D_{in})$
    $Max\_Number\_Of\_Shortest\_Paths\_Per\_Demand(K_{max})$,
    $Eclipse\_Nodes\_List(ENL))$
if $x_{initial} == x_{goal}$ then
    $return\ x_{goal}.Nodes\_list$
end if
$Demands\_shortest\_paths\_dictionary(DSPD) = \{\ ,\}$
for i=1 to size($D_{in}$) do
    if $K_{max} <= 0$ then
        $SP\_Of\_Demand = get\_all\_shortest\_paths\_(D_i)$
    else
        $SP\_Of\_Demand = get\_K_{max}\_of\_shortest\_paths\_(D_i)$
    end if
    $DSPD.add(D_i, SP\_Of\_Demand)$
end for
$Initialize\ Queue(Q)\ by\ (x_{initial}, cost = 0)$
while $true$ do
    $x = dequeue\_lowest\_cost\_state(Q)$
    if $size(Q) == 0$ then
        $Return\ No\ Solution$
    end if
    if $x == x_{goal}$ then
        $return\ x.Nodes\_list$
    end if
    for i=1 to size($DSPD.Keys$) do
        $Demand = DSPD.Keys(i)$
        if $Demand \in x.Demand_{List}$ then
            $Continue$
        end if
        $List\_SP = DSPD.Value(Demand)$
        for j=1 to size($List\_SP$) do
            $x_{new} = add(List\_SP(j), x)$
            $Old\_Len = length(x.Nodes\_list \cap ENL)$
            $New\_Len = length(x_{new}.Nodes\_list \cap ENL)$
            $cost = New\_Len - Old\_Len + x.cost$
            $Q.enqueue(x_{new}, cost)$
        end for
    end for
end while
```

V. EXPERIMENTAL EVALUATION

In this section we evaluate the proposed heuristics using simulations. First we describe our experimental setup. Then we quantify the energy saving on LEO satellite network with different traffic patterns. We aslo measure the impact of this energy-efficient heuristics in route length.

A. Exprimental Setup

Cplex version 12.6 was used to solve the ILP formulations, while a custom simulator is written in Matlab to implement the heuristics. In order to assess the performance of the proposed heuristics, we consider LEO satellite constellation which is described in the previous paper. The goal is to show that, for a real traffic demand, it is possible to power off satellite nodes, and still guarantee full connectivity between source and destination satellites, while enforcing that the network link utilization stays smaller than a quality of service (QoS) threshold. We tested the performance of the proposed heuristics in both off-peak traffic scenario and regular traffic scenario. The link capacity vaule is 155 Mbps. The link utilization α set to 0.5. Each time a different value of the “total offered traffic” is chosen. Finally, the power consumption for satellite node located in eclipse and sun is supposed to be 1000 Watt and 0 Watt, respectively, and the power consumption for satellite link located in eclipse and sun is supposed to be 50 Watt and 0 Watt, respectively. The optemization was performed on a Linux PC with eight 3.4 GHz CPU and 16 GB RAM.

Because CPLEX takes several hours to find an optimal solution even for the smallest network (e.g., Atlanta network, 15 nodes and 22 links), we force CPLEX to stop after three hours of execution. taking the fact that, limit the computation time to 300 seconds only in CPLEX can obtain about 96% of the optimal power saving [15]. Finally, we set the maximum number of shortest paths per demand ($k_{max} = 3$) to decrease the searching time in both heuristics.

B. Energy Saving

We start by assessing the amount of power saving that different algorithms obtain for LEO satellite topology and different traffic matrices.

Given the LEO topology, we generate 10 different traffic matrices, both the optimal formulation and the heuristics are run to evaluate the percentage of power savings that can be obtained, computed as

$$Saving = 100 \left(1 - \frac{\sum_{(u,v) \in E} p_{uv} x_{uv} + \sum_{u \in V} p_u y_u}{P_{tot}} \right) \tag{6}$$

Fig. 2 details the percentage of power saving obtained from LEO topology with ten different traffic matrices. Bars report mean values, while the error bars show the minimum and maximum power saving computed over ten independent runs. The maximum error of power saving is below 10 % with 95 % confidence for all algorithms. In this case, we report results considering all the heuristics. CPLEX is able to guarantee the largest amount of power saving (as expected), while MENH

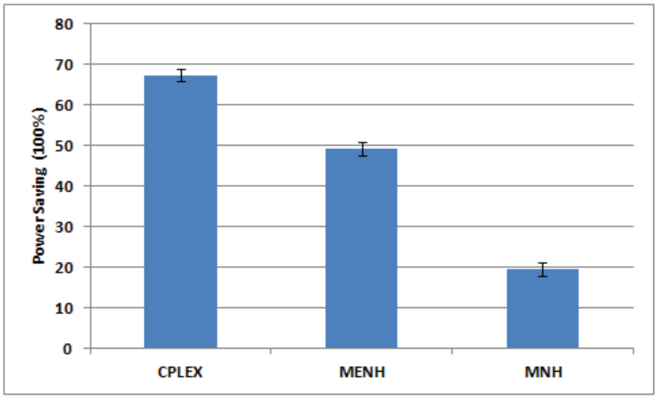

Figure 2: Power saving considering different algorithms. LEO topology with ten different traffic matrices is considered

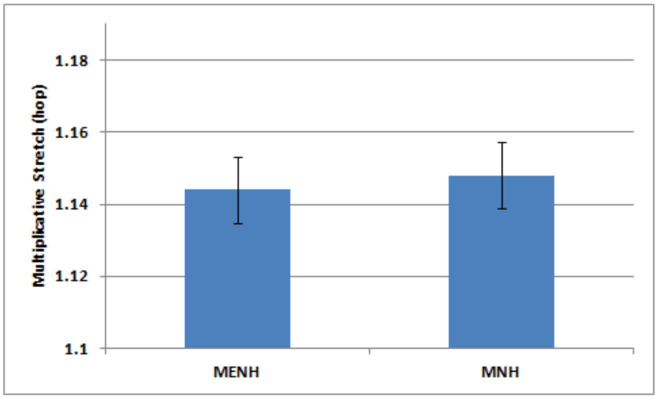

Figure 3: Impact of the Energy Efficiency Heuristics on the Routing Length (Average Multiplicative Stretch Factor)

leads to a much better power saving over the LEO constellation when compared to MNH. This is to be expected since MENH adapts to the position of the satellite while MENH does not adapts the position of the satellites. In brief, MENH algorithm favored to turn on satellite nodes exposed to the sun (powered by solar panel).

As power saving increase in eclipsed satellite nodes, the depth of discharge will be decreased in the satellites batteries. This leads to increase the availability of LEO satellite constellations. Moreover, more energy efficient network architectures would allow LEO network deployments in less developed parts of the words.

C. Stretch Factor

We believe that the LEO network operators will implement energy efficient techniques only if the impact on the other parameters is limited. We discuss here the impact on the route lengths of our proposed heuristics, because, it is the most important parameter in LEO topology.

When turning-off some network elements, we save some energy but, at the same time we route on longer paths. The **multiplicative stretch** is defined as the ratio between the average route length in the new topology divided by the average route length with the original topology (using all network nodes and links).

We compute the average multiplicative stretch (which also gives an idea of the delay) between our heuristics and the shortest path (SP) routing in the original topology. We used the same 10 traffic matrices generated in the previous section to compute the average multiplicative stretch.

Results for the LEO topology with the 10 traffic matrices are given in Fig. 3, we see that the impact on the route lengths is limited. E.g., the average increase is 1.15 % for both heuristics approximately. In general, the increase for the extreme cases spans from 13% to 18% and in average 15%. Fig. 3 shows that both heuristics lead approximately to the same multiplicative stretch. This is to be expected since MNH and MENH use the same technique to serve all the input demands. However, MENH heuristic significantly outperforms the FGH heuristic with regards to power savings. Because, MENH turn on more sun satellite nodes compare to MNH algorithm.

VI. CONCLUSION

Survivability and energy efficiency are a necessity in LEO satellite networks worldwide. However, when traditionally implemented they often counteract each other. This paper developed and evaluated techniques that save energy and improve availability in LEO satellite network by selectively powering down networks elements (nodes and links). We have first formulated the problem using an ILP formulation, showing that the problem falls in the class of NP-hard. Then, we developed several easy to implement heuristics. Evaluations based on LEO topology and real traffic matrices show that, our heuristics are able to achieve considerable power savings with minor impact on the route length. As our solution can be implemented using existing hardware technologies over the LEO topology, we believe that our heuristics will be of significant interest to the constellation operators community.

REFERENCES

[1] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 19–26.

[2] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing isp network energy cost: formulation and solutions," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 2, pp. 463–476, 2012.

[3] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*. ACM, 2010, pp. 29–34.

[4] A. C. Fu, E. Modiano, and J. N. Tsitsiklis, "'optimal energy allocation and admission control for communications satellites," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 3, pp. 488–500, 2003.

[5] M. Hussein, G. Jakllari, and B. Paillassa, "On Routing for Extending Satellite Service Life in LEO Satellite Networks (regular paper)," in *IEEE Global Communications Conference (GLOBECOM), Austin, TX, USA, 08/12/2014-12/12/2014*, and, Ed. http://www.ieee.org/: IEEE, dÃľcembre 2014.

[6] F. Ganhão, L. Bernardo, R. Dinis, G. Barros, E. Santos, A. Furtado, R. Oliveira, and P. Pinto, "Energy-efficient qos provisioning in demand assigned satellite ndma schemes," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*. IEEE, 2012, pp. 1–8.

[7] F. Alagoz, O. Korcak, and A. Jamalipour, "Exploring the routing strategies in next-generation satellite networks," *IEEE Wireless Communications*, vol. 14, no. 3, pp. 79–88, 2007.

[8] W. Jiang and P. Zong, "A discrete-time traffic and topology adaptive routing algorithm for leo satellite networks," *Int'l J. of Communications, Network and System Sciences*, vol. 4, p. 42, 2011.

[9] C. Chen and E. Ekici, "A routing protocol for hierarchical leo/meo satellite ip networks," *Wireless Networks*, vol. 11, no. 4, pp. 507–521, 2005.

[10] O. Korfak and F. Alagoz, "Deflection routing over prioritized intersatellite links in leo satellite networks," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 2. IEEE, 2005, pp. 2485–2490.

[11] I. Ghamlouche, T. G. Crainic, and M. Gendreau, "Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design," *Operations research*, vol. 51, no. 4, pp. 655–667, 2003.

[12] C. R. O. Longo, S. L. Rickman, and L. B. J. S. Center, *Method for the calculation of spacecraft umbra and penumbra shadow terminator points*. National Aeronautics and Space Administration, 1995.

[13] G. Ning and B. N. Popov, "Cycle life modeling of lithium-ion batteries," *Journal of The Electrochemical Society*, vol. 151, no. 10, pp. A1584–A1591, 2004. [Online]. Available: http://jes.ecsdl.org/content/151/10/A1584.abstract

[14] J.-W. Lee, Y. K. Anguchamy, and B. N. Popov, "Simulation of chargeï£¡discharge cycling of lithium-ion batteries under low-earth-orbit conditions," *Journal of Power Sources*, vol. 162, pp. 1395–1400, 2006.

[15] M. Zhang, C. Yi, B. Liu, and B. Zhang, "Greente: Power-aware traffic engineering," in *Network Protocols (ICNP), 2010 18th IEEE International Conference on*. IEEE, 2010, pp. 21–30.