Green distributed algorithm for energy saving in IP wired networks using sleep scheduling

Mohammed Hussein, Wisam Alabbasi, Ahmad Alsadeh
Department of Electrical and Computer Engineering, Birzeit University, Palestine

Article Info	ABSTRACT
	Energy saving has become a critical issue and a great challenge in the past few decades, and a great effort as well is being made to reduce consumed energy. The Internet forms a major source for energy consumption. Therefore, in this work we propose an algorithm for energy saving in distributed backbone networks, the reduced energy consumption (RedCon) algorithm. In this paper, we introduce a new version for saving energy on the Internet by switching off underutilized links and switching on idle links when the network is overloaded in a distributed manner over the network nodes based on LSA messages and without any knowledge of the traffic matrix. Our algorithm is more accurate and outperforms other algorithms with its time checks and advanced learning algorithm.

Corresponding Author:

Mohammed Hussein
Department of Electrical and Computer Engineering
Birzeit University
Ramallah, Palestine
Email: mhussein@birzeit.edu

1. INTRODUCTION

Energy is a source of life. It is needed to maintain development and progress of human work and achievements. Wasting energy when it can be saved is an obstacle in the way of development, taking into consideration that energy we use is not renewable, it is gone when used. Accordingly, we must try to save energy in every aspect we can, and this is the justification of this paper. In the past few decades power consumption has been a major worldwide issue taking too much of scientists and researchers time and effort to reduce the amount of consumed power due to its huge effects from global CO_2 omissions to the great waste of energy [1]-[4].

There is no doubt that the Internet forms a major source of power consumption due to its huge infrastructure of routing and switching devices, and customer devices in addition to its exponential growth in the last years. As known network traffic load differs from time to time, resulting in peak hours in which the network is overloaded and highly congested and off-peak hours in which the network is not congested at all and have the ability to power off some network devices and links, while preserving connectivity and reliability. Therefore, the main idea here is to keep the network connectivity with no packet loss and with the least amount of energy consumption [5]-[7].

The algorithm proposed in this work for energy saving on the internet, reduce energy consumption by switching off underutilized links and switching on idle links when the network is overloaded in distributed networks based on link state advertisement messages (LSA messages) with no network previous knowledge. In the introduced algorithm, the network topology and congestion information are identified based on LSA messages. Then the decision of choosing a certain configuration as the state of the link among a set of

configurations is made based on the current status of the node and link and its history. Meaning that if a previous decision has led to a network congestion or disconnection at a certain time it will have different penalty values computed based on its impact. The learning algorithm we used in this approach is double deep Q-learning (DQN) algorithm [8], [9].

This paper is organized as shown in: section 2 provides a general overview of energy saving in IP networks algorithms, section 3 discusses our proposed algorithm RedCon approach, and section 4 illustrates RedCon implementation, section 5 explains the evaluation methodology, features, collection, and results. Finally, section 6 summarizes the conclusions.

2.   RELATED WORKS

Different approaches appear in the literature to solve the problem of power consumption on the internet [10]-[12]. These approaches are classified into two main categories: techniques that adapt the consumption relatively to the traffic load by modifying the design of hardware devices, while the second category involves coordination among hardware devices by putting some network devices and links into sleep mode. In this paper, we are interested in the second category that comes with different implementations and algorithms. Sleep mode algorithms differ in the way they choose the optimal set of devices to be powered off, in their centralized or distributed nature, and their knowledge of the traffic matrix. These algorithms involve network topology pruning through powering off some network components in a proportion to the traffic load during off-peak hours without affecting the network connectivity and performance. A distributed algorithm GRiDA, which put links in IP-networks into sleep mode based on OSPF protocol to limit the amount of shared information and reduce the algorithm complexity. GRiDA is a blind traffic algorithm depends on the current load and the history of past decisions to power off links. The algorithm was tested on 3 different real scenarios: metropolitan segment network, Geant network, and European-wide network. Results show that GRiDA outperforms other centralized solutions with perfect knowledge of the traffic matrix.

Different formulation of the same problem is proposed in [2], a method for energy saving in IP networks by powering of less used links in the network, while keeping it connected, the resulted network topology quality is quantified using a quality metric called the Adequacy Index ADI and two polynomial time heuristics called Algebraic based algorithm for frugality (ABStAIn), which depends on the Algebraic connectivity concept to decide the importance of the link, and centrality based algorithm for frugality (CuTBAck), which depends on the link between centrality to decide how important the link is. These two instantiations are used to create the frugal topologies till a certain ADI threshold is reached. Furthermore, in the proposed approach we have the ability to enable the energy saving mode during off-peak hours and disable it during peak hours. The researchers used 3 real network topologies in the experiment and then compared ABStAIn and CuTBAck results with the optimal solution computed using the MATLAB "OPTI" tool, Benchmark-MLU and Least-Flow solutions. The results showed that the proposed approach was able to switch off about 25% of the links in the network and 80% as many links as a state of the art polynomial time solution, also with the same effectiveness as an exponential time traffic based solution at creating frugal topologies. And in Cuomo et al. [13] a new algorithm energy saving based on Algebraic connectivity (ESACON) is proposed, it leverages the Algebraic connectivity of the graph theory in order to identify the links that can be powered off during off-Peak hours while keeping the network connectivity above a fixed threshold with the aim of reducing the power consumption in ISP networks. The proposed algorithm can be implemented in centralized and distributed scenarios as well. The algorithm performance results were compared to the performance results of a topology-aware solution and a traffic aware-solution; the comparison results have shown that the proposed algorithm outperforms the topology-aware solution and has a comparable performance with the traffic-aware solution, noting that ESACON does not require the knowledge of the actual traffic.

A new approach was proposed in Cuomo et al. [14] for energy saving in IP networks by powering off less used links in the network based on the occurrences of links and nodes using the classical Dijkstra algorithm; this approach is called ESOL, Energy Saving based on Occurrence of Links. ESOL comes in four versions (b-ESOL, f-ESOL, (f + b)-ESOL, (f x 2)-ESOL) that present different tradeoffs such as complexity, execution time, and efficiency in powering off less used links. Finally, the performance results of these four algorithms were analyzed by applying them to real IP autonomous systems topologies and to other solutions. The results showed that in the three used topologies the percentage of powered off links is between 60%-90%, while still guaranteeing the network connectivity, b-ESOL is the most effective but slow, f-ESOL is the fastest with a less percentage of powered off links than b-ESOL, a combination of b-ESOL and f-ESOL increases the number of powered off links, and the repetition of f-ESOL presents a good approximation of b-ESOL with very reduced time.

In Cuomo *et al.* [15] the authors proposed a traffic-unaware heuristic called energy saving based on topology control (ESTOP), which reduces energy consumption on the Internet by identifying poorly used router line cards and putting them into sleep mode based on algebraic connectivity property and link betweeness parameter, while keeping the primary topological characteristics of the network. Experiment tests were done on two different sets of real ISP networks with many different topologies and then the performance results were compared to other topology-aware and traffic-aware solutions. The experimental results show that ESTOP outperforms other topology-aware solutions in terms of traffic with improved network utilization. Furthermore, ESTOP has a comparable performance with the traffic-aware solutions.

Moreover, in Cuomo *et al.* [16] the authors have introduced a topology oriented approach for energy saving on the Internet by identifying underutilized router line cards through Internet Service Providers graph modelling. In this approach the authors used ESTOP as a traffic unaware method for topology control. The proposed approach has proved its efficiency by outperforming other existing topology aware solutions.

Finally, Hussein *et al.* [17] and Hussein *et al.* [18], we proposed a novel approaches to reduce energy consumption in LEO satellite networks. In our proposed algorithm, we follow a different approach: we propose a novel distributed algorithm, called RedCon, to put into sleep mode links in wired network. Our solution is distributed among the nodes to; i) limit the amount of signaling information, ii) avoid explicit coordination among nodes, and iii) reduce the problem complexity. Contrary to previous works, we assume that nodes do not know the traffic matrix, since the learning algorithm we used in this approaches double deep Q-learning (DQN) algorithm. RedCon is able to react both to traffic variations and link/node failures.

3. APPROACH DESCRIPTION

In this section we describe the proposed algorithm main concept, which depends on switching off under-loaded links and switching on idle links whenever the network state is overloaded and more capacity is required to preserve network connectivity and reliability. This process is distributed among network nodes, which take local decisions at random time intervals based on the network state. Therefore, the traffic load and power consumption are already known to the network nodes. The network topology and congestion information are identified via the link state advertisement messages. Once the network links are overloaded, LSA messages report a violation to the network nodes. The local decision of each node corresponds to choosing a specific configuration from a set of all possible configurations for that node, where the configuration represents the state of each link whether on or off. The status of the node is identified by observing the status of all links associated to that node, which may take three possible values: off, normal, and overloaded according to a certain threshold value. And the utility function is calculated by computing the power consumption of on-links and the penalty associated to each configuration based on the node status and history. Meaning that if a configuration choice has previously led to network congestion or disconnection reported by an LSA message, it is not applied and its penalty is updated by β value. The process of selecting the configuration with the least power consumption while preserving the network connectivity is solved using the double DQN algorithm, since the utility function is updated by learning based on the configuration penalty. The researchers have proposed the double DQN algorithm that uses the existing architecture and deep neural network, and it has shown to be more stable and reliable than other learning algorithms.

If a configuration choice is previously followed by a network congestion or disconnection, the penalty function is updated by adding β value. Otherwise, the penalty function is updated by the following criteria: if entering a configuration choice when having a certain status has previously led to a network congestion or disconnection reported by an LSA message, the LSA time will be checked first if it occurs at the same time (same day of week, same day of month and month, and same hour) as in Figure 1. Otherwise, the penalty function is updated by a multiplicative value of δ as shown in (1). First of all, the penalty function for each possible configuration is initialized according to the criteria in (2) and it is updated by learning based on history and status.

$$P(J, S) = P(J, S) * \delta \ \forall J \in Kn \tag{1}$$

$$\theta(S, K) = \begin{cases} 0, S = OFF \ \cup \ K = 1 \\ 1/d, S = Normal \ \cap \ K = 0 \\ \varepsilon/d, \ S = Overloaded \ \cap \ K = 0 \end{cases}$$

Therefore, the cost associated to that configuration choice is according to the following criteria:
−   If a configuration choice is previously followed by a network congestion or disconnection at the same day of week, day of month and month, and same hour, the penalty function is updated as shown in (3).

- If a configuration choice is previously followed by a network congestion or disconnection with two similar time options, the penalty function is updated as shown in (4).
- If a configuration choice is previously followed by a network congestion or disconnection at the same day of week, or the same day of month and month, or the same hour, the penalty function is updated as shown in (5).
- If a configuration choice is previously followed by a network congestion or disconnection at two completely different times, the penalty function is updated as shown in (6).
- Otherwise, the penalty function is updated by a multiplicative value of δ as shown in (1).

$$P(K, S) = P(K, S) + 4\beta \tag{2}$$

$$P(K, S) = P(K, S) + 3\beta \tag{3}$$

$$P(K, S) = P(K, S) + 2\beta \tag{4}$$

$$P(K, S) = P(K, S) + \beta \tag{5}$$

```
Input: K, Kold, Sold, p, date
Output: K, p
Counter=0
if(Check Day_of_Week==True):
        Counter++
if(Check Day_of_Month==True && Check Month==True):
        Counter++
if(Check Hour==True):
        Counter++
if to_be_checked == True:
        if LSA==OK:
                for j in k:
                        p(j, Sold) = p(j, Sold) * δ
        else:
                if(counter==3):
                        p(j, Sold) = p(j, Sold) + 4β
                elif (counter==2):
                        p(j, Sold) = p(j, Sold) + 3β
                elif(counter==1):
                        p(j, Sold) = p(j, Sold) + 2β
                else:
                        p(j, Sold) = p(j, Sold) + β
                K= Kold
        to_be_checked = False
```

Figure 1. Proposed algorithm pseudo-code of the LSA arrival

4. SCENARIO DESCRIPTION

We used three different scenarios to test the performance and feasibility of our proposed method. The power model and network scenarios are discussed in the following subsections.

4.1. The power and traffic model

The power model used in our study is the same power model used in [19]. With a network topology composed of below devices and their power consumption default values used in our research, without considering air conditioning responsible for nearly 50% of total energy consumption:
- 372 routers.
- 8 core nodes, with 10 KW mean power consumption and 9.64% of total node power.
- 52 backbone nodes, with 3 KW mean power consumption and 19.03% of total node power.
- 52 metro nodes, with 1 KW mean power consumption and 6.32% of total node power.

−   260 feeders, with 2 KW mean power consumption and 65.19% of total node power.
−   718 links cardinalities.

Since our method depend on links utilization for energy saving via switching them on or off based on their current status, we are going to consider the power consumption saving related to the links and the amplifiers used through these links. Therefore, we are using below default values in our power consumption model, same as used in [19]:

−   Link capacity is 10 Gbps, Lcd = 10 Gbps.
−   An amplifier is used every 70 Km, Mld = 70 Km.
−   Port consumption is 50 W for each link, Cp = 50 W.
−   Amplifier consumption is 1 KW for each link Ca = 1 KW.
−   constant traffic requests on fixed time intervals of 30 minutes. The traffic matrices are obtained from direct measures were applicable or calculated from a single traffic matrix with an artificial traffic profile, Δ TM [min] = 30 or greater

Therefore, the power consumption Pcon in our model for each link L, capacity Lc, and Length Ml is computed as:

$$Pcon = \left\lceil \frac{Lc}{Lcd} \right\rceil \left(\left\lfloor \frac{Ml}{Mld} \right\rfloor Ca + 2Cp \right).$$

4.2. The network scenarios
4.2.1. ISP 1

The first testing scenario is an access/metropolitan segment of a traditional telecom operator network. This network is represented in Figure 2, where the access nodes are the blue labeled nodes with 1 to 8, they cannot be switched off and the represent the source and destination for the requests. Other nodes, labeled with 9 to 21 are the transit nodes. They are responsible for traffic transportation, can be switched off or on. Whereas the T node is the collection point, providing access at the transport network and the Internet level. Links and amplifiers are demonstrated as black squares where they exist [20]. The same traffic matrix used in [20], has been used in this work with max. Link utilization less than 70% with total of 47 traffic matrices and illustrated in Figure 2 with the label ISP 1.
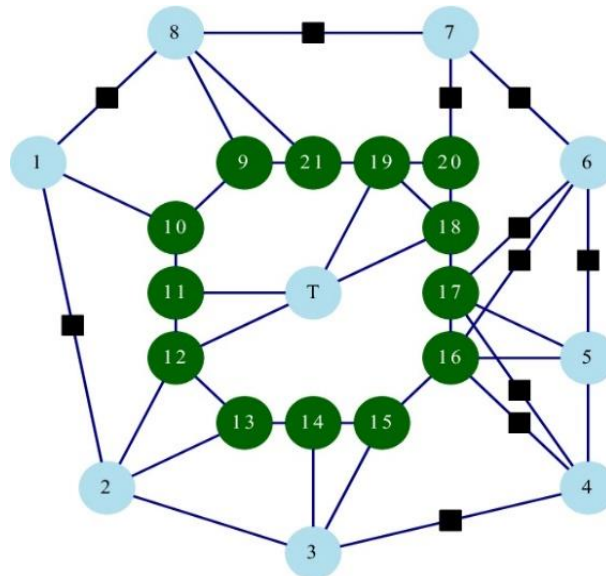

Figure 2. A telecom operator network topology: ISP 1

4.2.2. Geant

The original Geant network topology is considered in this work as illustrated in Figure 3 with blue nodes representing network nodes used as source and destination nodes, and the amplifiers are demonstrated as black squares where they exist with their counts. 48 actual traffic matrices for 05/05/2005 out of publicly available matrices are used in this work, illustrated in Figure 4 with the label Geant [21].
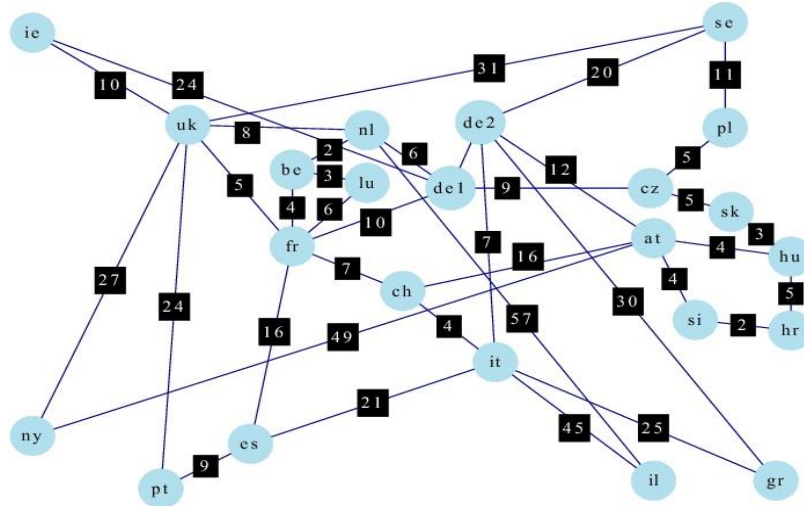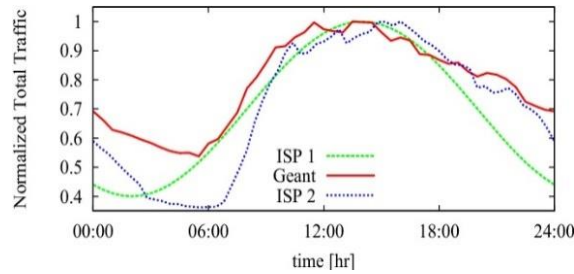
Figure 3. Geant network topology

Figure 4. Total traffic load and time variation

4.2.3. ISP 2

In this scenario, we considered a topology similar to real ISPs topology with hierarchical network illustrated in Figure 5 with many levels, Core level, Backbone level, Access level, Metro level, and the Internet level with 373 nodes and illustrated in Figure 5 with the label ISP 2 [19].
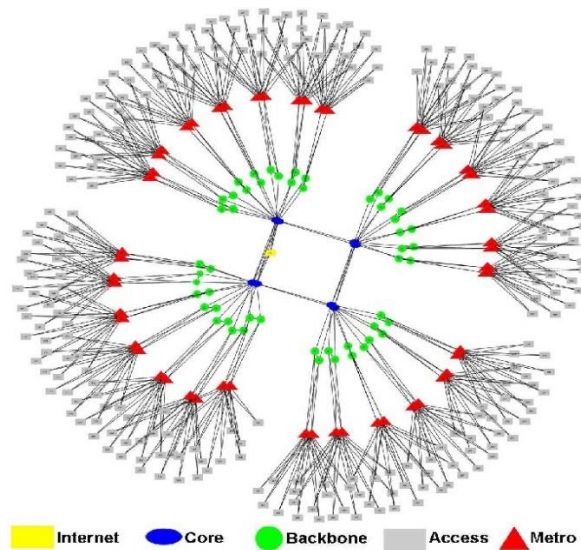

Figure 5. ISP2 topology

4.3. Parameter setting

A new TM is considered every time interval. A randomly selected node is waken up to take a decision every random interval Δc, uniformly distributed between Δ LSA and Δc, max seconds. Time intervals must be chosen in order to have, on the one hand, at least one LSA occurrence between two consecutive decisions, and on the other hand, a significant number of decisions per node to allow algorithm convergence. On average, a single node takes a decision every Δc×N, where N is the number of nodes in the network.

Values for the parameters in the different simulation scenarios are summarized in Table 1. The number of nodes for the ISP 2 network is divided into two parts, as the first part (i.e., core, backbone, metro nodes) is running the RedCon algorithm, while the second part (i.e., access and Internet nodes) is not running the RedCon algorithm. Access nodes in the ISP 2 network are not directly connected among them, hence, every link is considered in the RedCon algorithm even if they are not running it.

Table 1. Simulation parameters in the three simulation scenarios

Parameter	ISP 1	Geant	ISP 2
Δ LSA [S]	5	5	2
Δ TM [min]	30	30	48
Δ c,Max [s]	25	25	9
N	22	23	112+261
δ	1.0	0.999	0.999
β	50	50	100
ε	50	50	50
ε	0.7	0.7	0.5
Choices/Node/Traffic Matrix	5.5	5.2	4.7

5.   PERFORMANCE EVALUATION

A custom simulator is written in MATLAB to implement the RedCon algorithm. Node choices, LSA arrivals and traffic matrices changes are the possible events. Also, the network statistics including, node configurations, link load and power consumption are stored in a log file. Unless otherwise specified, we simulate a time period of one week, by repeating the set of traffic matrices. In the following, we first analyze the transient behavior of the algorithm on the different scenarios. As a second step we consider the sensitivity of average performance metrics to parameter variations.

Basis for comparison: we compare RedCon with the optimal solution computed using CPLEX tool and a well-known traffic based solutions [22], [23]. To measure the value of a carefully designed solution we also compare to a solution that simply removes links at random (R). 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 300 seconds 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 [24], [25].

5.1. Experiment 1: transient analysis

The performance of RedCon algorithm has been evaluated on ISP1 topology, using δ = 1 to test the convergence of the proposed algorithm. Then compared the performance of RedCon by the optimal solution obtained by CEPLEX for the off-peak traffic. Moreover, we compare the proposed solution with the centralized least-flow (LF) [22] which is a greedy heuristic that iteratively selects the least loaded link in the network as a candidate for being switched off. In particular, for each given link, the link is switched off. Then traffic is routed over the residual capacity. If network connectivity and maximum link utilization constraints are met, link is definitively powered off. Else the link is left in on state. A full knowledge of the traffic matrix is assumed to route traffic on the residual network and check connectivity constraints. Figure 6 illustrates the power saving versus time of RedCon, LF, and R. The power here is represented as the percentage of saved power for a configuration where all links are turned on. As reported in the figure, after the first transient and since RedCon requires no network or traffic knowledge, it outperforms other algorithms with a constant higher power saving rate of 55%. RedCon converges quickly without any need to use the penalty function, noting that the δ value of 1 has largely affected the fast convergent to an optimal solution. In contrast, LF and R have to run whenever a change has occurred due to their centralized nature and dependency on the traffic matrix knowledge.

For more detailed analysis, Figure 7 illustrates the cumulative number of unaccepted changes due to network violations compared between the three algorithms of R, LF, and RedCon with three β values. As observed from the figure, greater values of β result in a less number of unaccepted changes and better

learning rate because the utility function takes the most aggressive action to reduce power consumption resulting in a high number of reconfigurations whenever the learning rate is low. But once the value of β is changed to greater values the number of unaccepted changes for RedCon steadily decrease. While R and LF have a constantly increased number of reconfigurations, which were considered as the number of times a certain link is left powered on in different heuristics, since these algorithms are centralized and need network changes. The performance evaluation has considered all the values mentioned in Table 1 with the same reported results.

From the results mentioned above, we can conclude that RedCon learns from network failures with a limited number of reconfigurations. Once a network failure has been detected the algorithm turns the links on, and then starts to turn the links off till it reaches a stable network configuration with an optimal power consumption and least number of unaccepted changes.
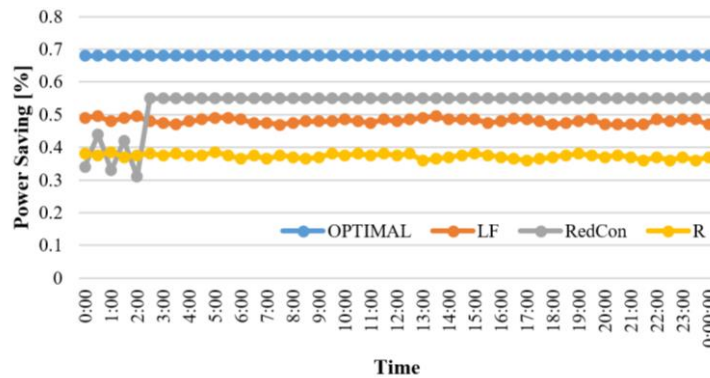

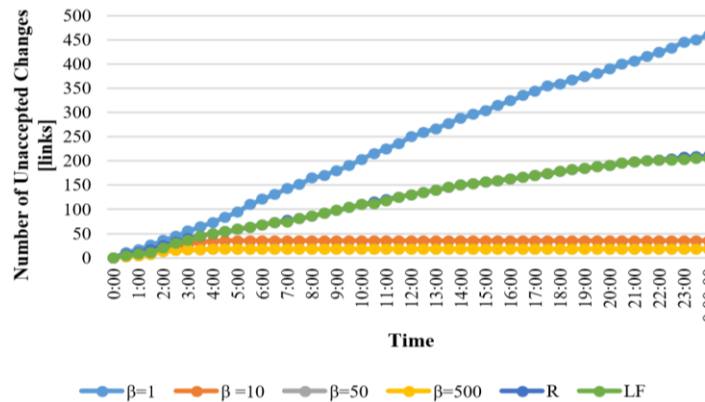Figure 4. Power saving versus time, considering different algorithms: ISP 1

Figure 7. Cumulative number of unaccepted changes for different β: ISP 1

5.2. Experiment 2: complex networks

The performance of RedCon algorithm has been evaluated on Geant topology, using δ=0.998 so that the proposed algorithm does not converge, instead it keeps adjusting the power saving to the actual network traffic. Then compared the performance of RedCon by the optimal solution obtained by CEPLEX for the off-peak traffic, LF, and R.

Figure 8 illustrates the power saving versus time of RedCon, LF, and R. As reported in the figure, the proposed algorithm outperforms other algorithms with a constant higher power saving rate. Also, the cumulative number of unaccepted changes due to network violations for RedCon is much lower than it is for R and LF algorithms as seen in Figure 9.

Figure 10 reports the algorithm comparison in terms of power saving. Interestingly, for all algorithms savings present a strong day-night trend. In particular, more power saving is possible when the network is lightly loaded, i.e., during night. In this case, RedCon is able to save an amount of power more than centralized algorithms, but without requiring the knowledge of the traffic matrix.
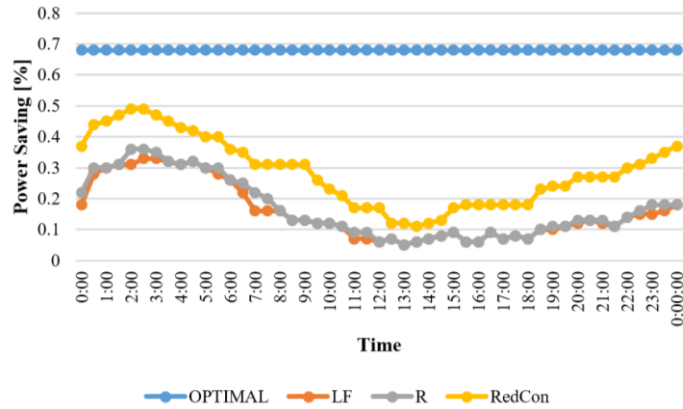
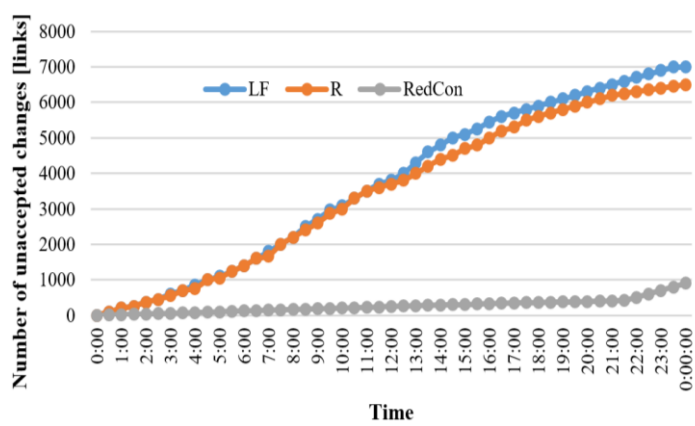Figure 8. Power saving versus time, considering different algorithms: Geant

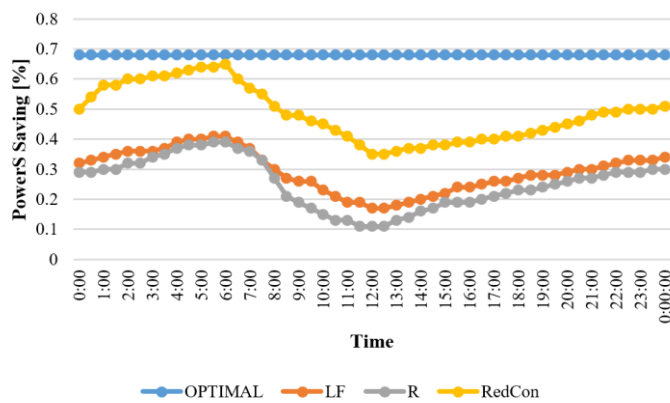Figure 9. Cumulative number of unaccepted changes: Geant

Figure 10. Power saving versus time, considering different algorithms: ISP 2

6.   CONCLUSION

We presented RedCon, a distributed heuristic for creating frugal IP networks. The new heuristics are based on a reinforcement learning technique that requires only the exchange of periodic link state advertisements in the network. Using topologies, a from 3 real networks–ISP 1, Geant and ISP 2- we showed that RedCon is able to switch off about 50% of the links in the network. We believe RedCon gives network operators a new option for managing the energy consumption of their infrastructure.

REFERENCES

[1] J. Lei, S. Deng, Z. Lu, Y. He, and X. Gao, "Energy-saving traffic scheduling in backbone networks with software-defined networks," *Cluster Computing*, vol. 24, pp. 279-292, 2021, doi: 10.1007/s10586-020-03102-5.

[2] M. Hussein, G. Jakllari and B. Paillassa, "Frugal topologies for saving energy in IP networks," *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, 2015, pp. 303-311, doi: 10.1109/LCN.2015.7366324.

[3] P. Modi, and M. Mathur, "Distributed WSN SHM with RF Wake-Adaptive Sleep Energy Saving & Artificial Intelligence Based Fatigue/Threat Perception with Early Warning System," *International Journal of New Technology and Research (IJNTR)*, vol. 5, no. 7, pp. 106-111, 2019.

[4] L. Pallavi, A. Jagam and B. T. Rao, "ERMO2 algorithm: an energy efficient mobility management in mobile cloud computing system for 5G heterogeneous networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, 1957-1967, 2019, doi: 10.11591/ijece.v9i3.pp1957-1967.

[5] F. D.-Zarandi, and Z. Movahedi, "A Dynamic traffic aware energy-efficient algorithm based on sleep-scheduling for autonomous systems," *Computing*, vol. 100, no. 6, pp. 645-665, 2018, doi: 10.1007/s00607-018-0589-6.

[6] M. Hussein and A. Abu-Issa, "Topology-aware approach for reducing power consumption in backbone networks," *International Journal of Networking and Virtual Organisations*, vol. 20, no. 4, pp. 356-376, 2019, doi: 10.1504/IJNVO.2019.100598.

[7] M. Hussein, Abu-Isa, T. Abdellatif, A. Iyad, and Abdalkarim, "Reducing power consumption in LEO satellite network," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2256-2265, 2021, 10.11591/ijece.v11i3.pp2256-2265.

[8] H. V. Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, vol. 30, no. 1, pp. 2094-2100.

[9] M. O. Osifeko, G. P. Hancke, and A. M. Abu-Mahfouz, "Artificial Intelligence Techniques for Cognitive Sensing in Future IoT: State-of-the-Art, Potentials, and Challenges," *Journal of Sensor and Actuator Networks*, vol. 9, no. 21, pp. 1-31, 2020, doi: 10.3390/jsan9020021.

[10] V.-P. Hoang, M.-H. Nguyen, T. Quan Do, Nhân Lê, and D. D. Bui, "A long range, energy efficient Internet of Things based drought monitoring system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1278-1287, 2020, doi: 10.11591/ijece.v10i2.pp1278-1287.

[11] B. Safaei, A. A. M. Salehi, M. Shirbeigi, A. M. H. Monazzah, and A. Ejlali, "Pedal: power-delay product objective function for internet of things applications," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 892-895, doi: 10.1145/3297280.3297565.

[12] C. Chen, D. Barrera and A. Perrig, "Modeling Data-Plane Power Consumption of Future Internet Architectures," *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, 2016, pp. 149-158, doi: 10.1109/CIC.2016.031.

[13] F. Cuomo, A. Abbagnale, A. Cianfrani and M. Polverini, "Keeping the connectivity and saving the energy in the internet," *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 319-324, doi: 10.1109/INFOCOMW.2011.5928831.

[14] F. Cuomo, A. Abbagnale and S. Papagna, "ESOL: Energy saving in the Internet based on Occurrence of Links in routing paths," *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2011, pp. 1-6, doi: 10.1109/WoWMoM.2011.5986481.

[15] F. Cuomo, A. Cianfrani, M. Plverini, and D. Mangione, "Network pruning for energy saving in the Internet," *Computer Networks*, vol. 56, no. 10, pp. 2355-2367, 2012, doi: 10.1016/j.comnet.2012.03.009.

[16] M. Hussein, G. Jakllari and B. Paillassa, "On routing for extending satellite service life in LEO satellite networks," *2014 IEEE Global Communications Conference*, 2014, pp. 2832-2837, doi: 10.1109/GLOCOM.2014.7037237.

[17] M. Hussein, G. Jakllari and B. Paillassa, "Network Pruning for Extending Satellite Service Life in LEO Satellite Constellations," *2015 IEEE International Conference on Data Science and Data Intensive Systems*, 2015, pp. 240-246, doi: 10.1109/DSDIS.2015.74.

[18] M. Hussein, A. Abu-Issa and I. Elayyan, "Location-aware load balancing routing protocol for LEO satellite networks," *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2018, pp. 1-7, doi: 10.1109/COMMNET.2018.8360244.

[19] L. Chiaraviglio, M. Mellia and F. Neri, "Energy-Aware Backbone Networks: A Case Study," *2009 IEEE International Conference on Communications Workshops*, 2009, pp. 1-5, doi: 10.1109/ICCW.2009.5208038.

[20] A. P. Bianzino, C. Chaudet, D. Rossi, J. Rougier and S. Moretti, "The Green-Game: Striking a balance between QoS and energy saving," *2011 23rd International Teletraffic Congress (ITC)*, 2011, pp. 262-269.

[21] "The GEANT network," [online]. Availabe: http://www.geant.net/, [Accessed 1 July 2020].

[22] L. Chiaraviglio, M. Mellia and F. Neri, "Minimizing ISP Network Energy Cost: Formulation and Solutions," in *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 463-476, 2012, doi: 10.1109/TNET.2011.2161487.

[23] J. Lei, S. Deng, Z. Lu, Y. He, and X. Gao, "Energy-saving traffic scheduling in backbone networks with software-defined networks," *Cluster Computing*, vol. 24, pp. 279-292, 2021, doi: 10.1007/s10586-020-03102-5.

[24] M. Zhang, C. Yi, B. Liu and B. Zhang, "GreenTE: Power-aware traffic engineering," *The 18th IEEE International Conference on Network Protocols*, 2010, pp. 21-30, doi: 10.1109/ICNP.2010.5762751.

[25] A. P. Bianzino, L. Chiaraviglio and M. Mellia, "GRiDA: A green distributed algorithm for backbone networks," *2011 IEEE Online Conference on Green Communications*, 2011, pp. 113-119, doi: 10.1109/GreenCom.2011.6082517.