

RESEARCH

Open Access



# On the performance of online and offline green path establishment techniques

Alejandro Ruiz-Rivera<sup>1</sup>, Kwan-Wu Chin<sup>1\*</sup>, Sieteng Soh<sup>2</sup> and Raad Raad<sup>1</sup>

## Abstract

To date, significant effort has gone into designing green traffic engineering (TE) techniques that consolidate traffic onto the minimal number of links/switches/routers during off-peak periods. However, little works exist that aim to green Multi-Protocol Label Switching (MPLS) capable networks. Critically, no work has studied the performance of green label switched paths (LSPs) establishment methods in terms of energy savings and acceptance rates. Henceforth, we add to the current state-of-the-art by studying green *online* and *offline* (LSP) establishment methods. Online methods rely only on past and current LSP requests while offline ones act as a theoretical benchmark whereby they also have available to them future LSP requests. We introduce a novel metric that takes into account both energy savings and acceptance rates. We also identify a new simpler heuristic that minimizes energy use by routing source–destination demands over paths that contain established links and require the fewest number of new links. Our evaluation of two offline and four online LSP establishment methods over the Abilene and AT&T topologies with random LSP setup requests show that energy savings beyond 20 % are achievable with LSP acceptance rates above 90 %.

**Keywords:** Green technologies; MPLS; Online and offline green LSP establishment methods; Traffic engineering

## 1 Introduction

The Climate Group Organization [1] reports that the Information and Communication Technology (ICT) industry accounts for up to 2 % of global carbon gas emissions. This is expected to increase further with the continued popularity of a number of services and technologies such as Internet protocol TV (IPTV) [2], Voice over IP (VoIP) [3], the Internet of Things (IoT) [4], and cloud computing [5]. In fact, global IP traffic is estimated to have a compound growth rate of 21 % from 2013 to 2018 [6], surpassing the zettabyte (1000 exabytes) threshold in 2016. As a result, analysis such as [7] forecasts that the global carbon footprint of telecommunication network devices will grow by 5 % each year between 2002 and 2020 due to the steady rise in electricity demand. The authors of [8] indicate that the United States of America (USA) alone uses 24 TWh per year, costing around \$24 billion annually.

In their seminal work, Gupta et al. [9] highlighted the need to reduce the energy consumption of the Internet. Their work inspired a number of research directions with the common aim of reducing the energy expenditure of routers/switches. In general, these works can be categorized as (i) *sleeping* [10], which aims to place sub-components of devices or devices themselves to sleep, (ii) *link adaptation* [11], which scales the energy consumption according to varying link utilization, (iii) *proxying* [8], which reduces network chatters by way of a proxy, and lastly (iv) *traffic engineering (TE)* [12–16], whereby traffic is routed across the minimal number of links and routers. We note that in general energy efficiency is a contemporary research issue that is of considerable interest to researchers. For example, a number of works have studied green TE approaches that consider application and data center characteristics; e.g., [17–20]. Another is designing energy aware protocols for next-generation networks such as Internet of Things [21, 22], vehicular networks [23, 24], and wireless sensor/mesh networks [25, 26].

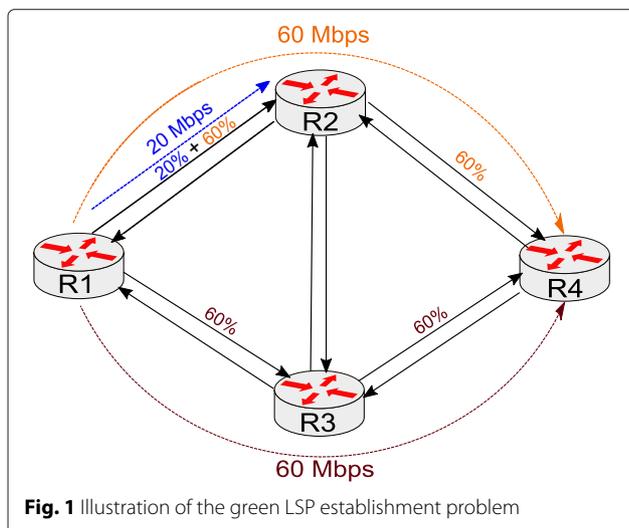
The methods investigated in this paper belong to the TE category. We focus on Multi-protocol Label Switching (MPLS) networks. In particular, green label switched

\*Correspondence: kwanwu@uow.edu.au

<sup>1</sup>School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, Australia  
Full list of author information is available at the end of the article

paths (LSPs) methods that use the fewest number of links. We note that establishing label switched paths (LSPs) to meet one or more QoS constraints or/and to maximize the number of admitted LSPs is a well-researched problem; see Section 2. However, there are only few works that are energy aware; see Section 2.

As an example, consider the MPLS network shown in Fig. 1. There are four label switching routers (LSRs) interconnected by directional links. Given a set of LSP requests, each with a source ( $s$ ) and destination ( $d$ ) address, and bandwidth demand ( $bw$ ), the problem is to establish the LSP for these requests that consume the least overall energy. Assume links have a capacity of 100 Mbps and currently they have zero utilization. Consider two LSP requests, each denoted as  $\langle s, d, bw \rangle$ , arriving at R1 in the following order:  $LSP_1 \langle R1, R2, 20 \rangle$ ,  $LSP_2 \langle R1, R4, 60 \rangle$ . Consider  $LSP_1$ . A solution may first assign it to link/path  $R1 - R2$  given that this is the shortest possible path, and the link can accommodate the requested demand. At this point, the utilization of link  $R1 - R2$  is 20 %. When  $LSP_2$  arrives, there are multiple paths to choose from, e.g.,  $[R1, R2, R4]$  and  $[R1, R3, R4]$ . We proceed to check whether any of these paths are able to accommodate the 60 Mb/s demand requested by  $LSP_2$ . If path  $[R1, R2, R4]$  is selected, the final utilization of the links  $R1 - R2$  and  $R2 - R4$  will be 80 and 60 %, respectively. If the path  $[R1, R3, R4]$  is selected, the utilization of links  $R1 - R3$  and  $R3 - R4$  will be 60 %. Therefore, both paths will be able to accommodate  $LSP_2$ . Now, if path  $[R1, R3, R4]$  is selected, all four routers need to be active, and a total of three active links, i.e.,  $R1 - R2$ ,  $R1 - R3$ , and  $R3 - R4$ , will be required to serve both LSP requests. However, if path  $[R1, R2, R4]$  is selected, link  $R1 - R2$  can be reused given that it is currently used to serve  $LSP_1$ . This means both LSP requests will only be served by a



total of three active routers, i.e., R1, R2, and R4, and use two active links; i.e.,  $R1 - R2$ , and  $R2 - R4$ . Selection of path  $[R1, R2, R4]$  for  $LSP_2$  will therefore accommodate the requested demand while minimizing network resource usage. From this simple example, we see that knowing the set of LSPs is critical to route selection and, consequently, the minimal set of active links that can support all these LSPs.

Current energy efficient LSP establishment methods, see Section 2, have the following limitations: i) they focus only on energy savings and neglect LSP acceptance rate, and ii) there is little or no work that provides a thorough comparison of online and offline LSP establishment methods. In light of these limitations, this paper makes the following contributions:

1. It compares the LSP acceptance rates of existing online and offline energy aware algorithms for establishing LSPs. We study (i) *offline* approaches, where the complete set of LSP setup requests is known in advance, and (ii) *online* approaches, where we only know the current and past LSP setup requests. In fact, this is the first study that compares all these approaches over the same topologies. Moreover, we propose an Integer Linear Program (ILP) formulation for the offline version of the problem.
2. This paper presents and studies the performance of two offline and four online heuristics: i) Offline Most Overlapped (*Offline-MO*) [27], a technique that aims to use paths that share the most links with past or/and future LSP requests, ii) Offline with Ratio (*Offline-R*), which is similar to *Offline-MO* but favors paths that require fewer number of new links, iii) Online Most Overlapped (*Online-MO*), which is similar to its offline counterpart, is an algorithm that uses paths that share the most links with already established links, iv) Online with Ratio (*Online-R*), which is similar to *Online-MO* but prefers paths that involve the minimal number of new links, v) Online Minimum Hops (*Online-MinH*) [28], an approach that gives priority to paths with a small number of hops, and vi) Online Random LSP (*Online-R-LSP*) [29] which selects paths randomly.
3. The paper also introduces and uses a ratio, called  $\rho$ , of the *percentage of shut down links* (PSL) and *LSP acceptance requests* (LAR) to quantify how well a method performs in terms of energy savings and its ability to accept new LSP requests. This metric  $\rho$  allows us to evaluate whether a solution that is able to accept a larger number of LSP requests also has significant energy savings. Extensive experiments involving well-known topologies such as Abilene and AT&T using varying traffic loads confirm that LSP

acceptance rates above 90 % are feasible with 20 % of links shut down.

The rest of the paper is structured as follows. Section 2 surveys related works. Section 3 outlines a formal definition of the problem. Section 4 introduces the proposed heuristics. Section 5 describes our research methodology and presents and discusses the evaluation results. Our conclusions are presented in Section 6.

## 2 Related work

This section is divided into two parts. The first part presents different green routing techniques. It then briefly describes works that aim to optimize the establishment of LSPs in conventional (non-energy sensitive) MPLS networks.

### 2.1 Green approaches

Current IP networks are designed to handle the *worst-case* scenario in terms of failures and traffic demands [10]; thus, not surprisingly, they are over-provisioned to handle peak traffic demands. On the contrary, many works have shown traffic to exhibit diurnal patterns [30]. Moreover, the utilization of backbone networks can be less than 30 % [8]. Consequently, given these observations, a number of researchers have proposed to power off network elements.

#### 2.1.1 LSP establishment methods

We start by reviewing works that deal with energy-efficient LSP establishment methods.

In [28], the authors propose an online routing algorithm, called Energy Efficient Multi-Constrained Routing Algorithm ( $E^2$ -MCRA), that aims to maximize LSP acceptance rates and minimize the number of active nodes and links while also considering additive QoS constraints. Its key ideas are to route a LSP request over paths with the minimum number of hops and instead of exhaustively exploring all paths, the algorithm searches only sub-graphs induced by active routers and links. In [27], the authors exploit pre-installed backup paths used to protect against failed links on a primary LSP. The proposed algorithm, called GBP, achieves energy saving by re-routing traffic from protected links onto these backup paths. In addition, long backup paths are avoided. Protected links are then shut down. References [31, 32] use an ILP to minimize the number of active links/routers during a given period of time while satisfying all traffic demands in every interval. The ILP's objective function considers the following factors: i) energy consumed by an active chassis, ii) energy consumed by line cards when powered on, and iii) the energy consumed by a chassis when transitioning from the off to on state. Coiro et al. in [33, 34] present Distributed and Adaptive Interface Switch-off for Internet

Energy Saving (DAISIES), an online algorithm that continuously monitors the load of each link and calculates a new set of link weights. Its aim is to reduce the number of links carrying traffic. The new link weights are then used by Dijkstra's algorithm to generate a new network topology, and links that are not part of this topology are switched off.

As we will elaborate in Section 5, the performance of  $E^2$ -MCRA is similar to that of an online heuristic that prefers to route an LSP over active links; see Section 5.3. Advantageously, the heuristic does not involve the expensive Look-Ahead and DFS phases of  $E^2$ -MCRA. In addition, one of the heuristics we investigated shares a key similarity to GBP whereby it avoids the use of long paths and aims to route over existing paths without having to establish new links.

#### 2.1.2 Green TE

To date, there are a number of green TE approaches. However, they do not consider LSPs. We briefly summarize a few key works. A modification of the Open Short Path First (OSPF) algorithm [35] is presented by Cianfrani et al. in [36]. They propose Energy-Aware Routing (EAR), whereby links that are not part of the shortest path tree (SPT) calculated by each router are switched off. In a different work, the authors of [12] formulated a mixed integer linear programming (MILP) problem to maximize the total power savings while minimizing the overall maximum link utilization (MLU). The main ideas are to select candidate paths with the minimum number of hops, shorter than the network diameter, or meet end-to-end delay constraints. In [37], the authors consider the problem of scheduling deadline constrained flows and their routing in a data center. Specifically, they formulated a convex optimization problem where the decision variables are to determine the transmission rate of a flow and its route such that a network's total energy expenditure is minimized. A key constraint is that the rate assigned to a flow must ensure all its data is transmitted within a given deadline.

In [38], routers are ranked according to one of the following criteria: i) random ii) degree iii) number of flows, or iv) number of active neighbors. They propose heuristics that either select the highest ranked router to be shut down or one with the highest power consumption. Wang et al. [39] aim to eliminate packet loss that occurs when consolidating traffic. They focus on IP networks that implement the Fast Reroute (FRR) NotVia technique [40], where for a given source–destination pair, there are multiple pre-computed paths that are used upon a failure.

In a different work, Bianzino et al. [13] propose an algorithm called Green Distributed Algorithm (GrIDA) that switches off links according to a link's current load and power consumption. Lin et al. [41] consider the QoS

of flows when switching off network elements. On the other hand, the Green Distributed Routing Protocol for Sleep Coordination (GDRP-PS) algorithm of [42] divides core routers into two sets: traditional routers (TRs) and power saving routers (PSRs). TRs are not allowed to sleep whereas PSRs are routers running GDRP-PS and, therefore, go to sleep whenever the traffic load is low. Before powering down, a PSR first checks whether the network remains connected if it shuts down. If so, it informs a coordinator it will enter sleep mode.

Athanasiou et al. [15] propose Energy-Aware Traffic Engineering (ETE), a distributed and offline algorithm that load-balances traffic while minimizing energy consumption and MLU. ETE is composed of the following algorithms: i) *Load Balancing* (LB). Each ingress node finds the amount of traffic destined to a given egress node that needs to be routed through each of its links in order to maximize link utilization; ii) *Energy Saving* (ES). Each ingress node takes an input traffic information from LB and uses it to calculate the minimum number of links required to carry said traffic. The authors of [43] propose heuristics to determine the network configuration that consumes the least amount of energy within a search space that contains all possible combination of nodes and active/sleeping links. To reduce the search space, the proposed heuristics generate all *patterns* for a given topology, where a *pattern* is a sub-graph of the original topology induced by a subset of sleeping links. In a different approach, the authors of [44] present Energy Aware TE (EATe), whereby traffic is spread across multiple paths. Cuomo et al. [45] introduce a topology aware algorithm called Energy Saving based on Algebraic CONnectivity (ESACON). This algorithm relies on the algebraic connectivity of a network. In particular, they rely on the fact that algebraic connectivity is a good indicator of a graph's robustness to node and link failures [46]. Finally, the authors of [47] present Energy Profile Aware Routing (EPAR), an energy aware TE approach that uses network equipment energy profiles (EPs) and builds on the Energy Aware Routing (EAR) algorithm proposed in [9]. EPAR accounts for the different EPs of devices and routes traffic along paths with energy-efficient components.

To the best of our knowledge, existing works thus far either do not consider LSPs, and for those that do, they have not considered the relationship between LSP acceptance rates and energy savings. This is critical because a given rule used to establish LSPs may yield a low acceptance rate; i.e., it blocks future LSPs from being admitted despite having large energy savings. In this regard, only [28] has considered both energy consumption and acceptance rate jointly. However, they only compared their proposed method to one other green LSP establishment approach and did not study the relationship between the remaining number of active links and the acceptance of

future LSP requests. We on the other hand consider six heuristics. Critically, we compare online against offline approaches and evaluate them using a novel metric that succinctly quantifies the advantage of an LSP establishment method in terms of energy saved and acceptance rate. Moreover, we present a new and simple heuristic that has comparable performance to the one presented in [28].

## 2.2 Non-green LSP establishment approaches

As noted in Section 1, there are many studies on increasing LSP acceptance rates by minimizing interference between LSPs. In addition, these studies also address the problem of establishing LSPs that satisfy one or more QoS constraints. Critically, the majority of these studies do not aim to conserve energy. In [48], Hong et al. propose Multiple Constraint-based Shortest Path First (MCSPF), which is based on the widely used Constraint Shortest Path First (CSPF) [49] algorithm. MCSPF aims to select LSPs that satisfy bandwidth and end-to-end delay constraints while maximizing LSP acceptance rates. In [50], the authors present a modification to the Wang-Crowcroft (WC) algorithm [51] in an effort to increase LSP acceptance rates by taking into account the order of arriving LSP requests. The resulting Wang-Crowcroft with Sorting (WCS) algorithm improves upon WC by reordering LSP requests according to their bandwidth and delay requirements. De Oliveira et al. [52] propose Stochastic Performance Comparison Routing Algorithm (SPeCRA), an algorithm that adaptively selects the best LSP establishment method, in terms of LSP acceptance rates, among a given set of candidate methods.

QoS routing approaches are discussed in [53, 54], namely, the Widest-Shortest Path (WSP), Shortest-Widest Path (SWP), and Shortest-Distance Path (SDP) algorithms. These algorithms establish LSPs between a given  $(s, d)$  pair by selecting paths that have sufficient bandwidth to satisfy the demand requested by the source  $s$ . WSP only considers the shortest path. If there are multiple equally good paths, WSP selects the one with the maximum bandwidth. SWP selects the path that contains the maximum available bandwidth. Lastly, SDP selects a path with the least cost, where cost is defined as the reciprocal of the available path bandwidth. The MPLS Adaptive Traffic Engineering (MATE) algorithm proposed by Elwalid et al. [55] aims to reduce network congestion by adaptively load balancing traffic across paths. The algorithm routes traffic using pre-established LSPs according to metrics such as packet delay, packet loss, or network utilization. In [56], the authors apply evolutionary-fuzzy strategy to predict the utilization level of links. This then allows routers to select a path for flows that are likely to accept the flow; i.e., their goal is to minimize the expected blocking probability.

In general, the aforementioned works optimize the allocation of LSPs based on different QoS constraints such as bandwidth, packet loss, and end-to-end delay, e.g., [48]. The order of LSP setup requests is also important when establishing paths [50]. Past works such as [52] have also shown the importance of minimizing interference in order to increase LSP acceptance rates. A key gap is that these methods have not taken energy consumption into account when selecting the optimum LSPs. Indeed, existing works assume the existence of redundant paths and nodes [55]. Green LSP establishment methods, however, have an opposite aim, whereby they seek the minimal number of nodes or links. Hence, this paper adds to the existing literature by analyzing and studying existing as well as new green LSP establishment methods and provides a comparison of their acceptance rates.

### 3 Problem description

Before defining the problem formally, we first introduce a number of notations. We model the network as a directed graph  $G(V, E)$ , with  $V$  being the set of  $|V|$  nodes, and  $E$  representing the set containing  $|E|$  edges. We denote the link between nodes  $i$  and  $j$  as  $e_{ij}$  or  $(i, j)$ . Each link has capacity  $c_{ij}$  and utilization  $u_{ij}$ . Let  $Q$  be the set of LSP establishment requests that arrive at the set of ingress routers  $I \subset V$ . Each LSP establishment request  $q \in Q$  is a tuple  $\langle s, d, bw \rangle$ , where  $s$  and  $d$  denote the source and destination of a request, and  $bw > 0$  is the corresponding requested bandwidth. We also define a function  $B(q)$  that returns the bandwidth of request  $q$ . Let  $P_q$  be the set of all simple paths that can be used to serve LSP request  $q \in Q$ . Specifically,  $P_q = \{p_q^1, p_q^2, \dots, p_q^{|P_q|}\}$  is a set of candidate paths for  $q$  sorted in increasing path length order. Each path  $p_q^k$  in  $P_q$  contains a set of  $|p_q^k|$  links, meaning  $p_q^k \subseteq E$ . We define the set of paths that use link  $(i, j)$  as  $P_{ij} = \{p_q^k \mid e_{ij} \in p_q^k\}$ , for all  $q \in Q$  and  $k = 1, 2, \dots, |P_q|$ . Hence, the total traffic over a given link  $(i, j)$  is  $B_{ij} = \sum_{p \in P_{ij}} B(p)$  and its link utilization is  $u_{ij} = B_{ij}/c_{ij}$ .

The problem at hand is as follows: given i) a MPLS network consisting of label switching routers (LSRs) and directional links with fixed capacity, ii) traffic demands described as a set of tuples  $\langle s, d, bw \rangle$ , which may be given a priori, i.e., offline, or in a real-time manner, i.e., online, the problem at hand is to minimize the overall energy consumption of the MPLS network by finding a set of LSPs that satisfy the given traffic demand of each request using the minimal number of links/routers. It is worth noting that in the online version of the described problem, the establishment of current LSPs affects the utilization of links/routers and hence may affect the acceptance of future LSP requests. The challenge is therefore to assign LSPs such that energy usage is reduced, while accommodating future traffic demands.

The offline version of the said problem can be modeled as an Integer Linear Program (ILP). Before outlining the objective function, we first introduce the decision variables and corresponding constraints. Let  $X_q^k \in \{0, 1\}$  be a binary variable that represents whether path  $k$  of request  $q$ , i.e.,  $p_q^k$ , is selected. As an example, consider the request  $q = \langle a, d, 10 \rangle$  in Fig. 2 and its two alternative paths  $p_q^1$  and  $p_q^2$ . Note that only one of the two paths can be selected, and thus, we set either  $(X_q^1 = 1$  and  $X_q^2 = 0)$  or  $(X_q^2 = 0$  and  $X_q^1 = 1)$ . In general we have the following constraint:

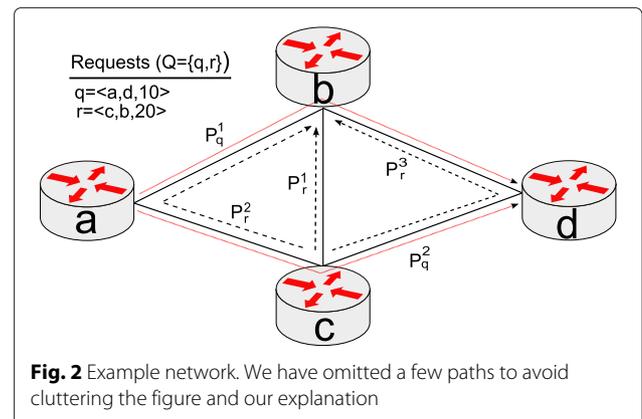
$$\sum_{k=1}^{|P_q|} X_q^k = 1, \quad \forall q \in Q \tag{1}$$

The next constraint ensures the load is within link capacity. Let  $\bar{T}_{ij}$  be the set of decision variables  $X_q^k$  that represent the paths that are using link  $e_{ij}$ ; i.e.,  $\bar{T}_{ij} = \{X_q^k \mid e_{ij} \in p_q^k\}$ , where  $k$  is an index corresponding to path  $k$  of request  $q$  that uses edge  $e_{ij}$ . For example, assuming request  $q = \langle a, d, 10 \rangle$  and  $r = \langle c, b, 20 \rangle$ , link  $(a, b)$  of Fig. 2 will have  $\bar{T}_{ab} = \{X_q^1, X_r^2\}$ . We thus have the following link capacity constraint,

$$\sum_{\chi \in \bar{T}_{ij}} \chi B(\chi) \leq c_{ij}, \quad \forall e_{ij} \in E \tag{2}$$

where the function  $B(\chi)$  returns the  $bw$  value associated with the path/request corresponding to decision variable  $\chi$ . As an example, consider Fig. 2 with demands  $q = \langle a, d, 10 \rangle$  and  $r = \langle c, b, 20 \rangle$ . The capacity constraint for link  $(a, b)$  is therefore  $X_q^1 \times 10 + X_r^2 \times 20 \leq 100$ , with  $c_{ab} = 100$ .

The final set of constraints ensure that a link or router remains active only if it is used by a path. Let  $X_{ij}$  and  $Y_v$  be binary variables that denote whether a link  $(i, j) \in E$  and router  $v \in V$  is active or inactive, respectively. In addition, we define a function  $F(v)$  that returns the  $X_{ij}$  of incident links on router  $v$ . To ensure that a link or router is active



only if there is a path that uses it, we have the following constraints:

$$X_{ij} \geq \bar{T}_{ij}, Y_v \geq F(\bar{v}) \quad (3)$$

where  $\geq$  represents the  $\geq$  operator executed component wise on the set  $\bar{T}_{ij}$  or  $F(\bar{v})$ ; referring to Fig. 2, link  $e_{ab}$  with decision variable  $X_{ab}$  will have the following constraints:  $X_{ab} \geq X_q^1$  and  $X_{ab} \geq X_r^2$ . That is, the decision variable  $X_{ab}$  is set to 1 only if  $X_q^1$  or  $X_r^2$  or both are set to 1.

Given constraints (1), (2), and (3), we seek to minimize the following objective function, which represents the total number of active links and routers:

$$\text{MIN} \sum_{e_{ij} \in E} X_{ij} + \sum_{v \in V} Y_v \quad (4)$$

Given that each router and link consumes a given amount of energy, minimizing the number of active routers and links would reduce the total energy consumed by a network. The objective function can also be adapted to include the specific power consumption of a NIC. We leave this as a future work.

The aforementioned offline version of the problem is solvable only for small networks due to the number of binary variables that grow exponentially with network size and demands. In particular, there could be an exponential number of paths that can be used for a given demand  $q$ . In fact, the offline version of our problem corresponds to the well-known multi-commodity minimum-cost flow (CMCF) problem and is therefore NP-complete; please refer to [57] for details. Henceforth, in the next section, we present different heuristics to address both online and offline versions of the formulated problem.

#### 4 Heuristics

In our discussion to follow, we will use Fig. 1 to describe our offline and online heuristics. Moreover, the LSP requests that arrive over time are shown in Table 1. The table also shows their respective  $k$  shortest paths. These paths can be calculated using Yen's  $k$ -Shortest paths algorithm [58].

In the following sections, we will describe the following heuristics in detail:

**Table 1** LSP setup requests and  $k$  paths shared by all implemented algorithms

$LSP_q$	$s$	$d$	$bw$	$k$ shortest paths		
1	R3	R1	14	[R3, R1]	[R3, R2, R1]	[R3, R4, R2, R1]
2	R4	R2	41	[R4, R2]	[R4, R3, R2]	[R4, R3, R1, R2]
3	R2	R4	40	[R2, R4]	[R2, R3, R4]	[R2, R1, R3, R4]

#### Offline

- Offline Most Overlapped (*Offline-MO*): aims to use paths that share the most links with past or future LSP requests.
- Offline with Ratio (*Offline-R*): same as *Offline-MO* but favoring paths that require the fewest number of new links.

#### Online

- Online Most Overlapped (*Online-MO*): aims to use paths that share the most links with already established LSPs.
- Online with Ratio (*Online-R*): same as *Online-MO* but prefers paths that involve the minimal number of new links.
- Online Minimum Hops (*Online-MinH*): gives priority to paths with a small number of hops.
- Online Random LSP (*Online-R-LSP*): selects paths randomly.

We remark that *Online-MinH* and *Online-R-LSP* have been considered in [28, 29], respectively. However, all other heuristics are new. Moreover, as noted in Section 2, no works have compared all these heuristics comprehensively. As we will show in Section 5, *Online-R* has the best performance in terms of energy saved and LSP acceptance rate.

#### 4.1 Offline approaches

Algorithm 1 presents a general overview of how offline heuristics are applied to each LSP request  $q \in Q$ . By definition, all these heuristics know in advance all LSP setup requests in  $Q$ , and their respective  $k$  shortest paths  $P_q$ . This means they can determine the best links to use or avoid by looking at past, current, and future LSP requests. Hence, the results obtained via offline heuristics constitute the best possible performance for any online heuristics. For the reader's convenience, Algorithm 1 also defines the variables used by the different *Heuristic(.)* functions.

For each arriving LSP request,  $q = \langle s, d, bw \rangle$ , the set of all shortest  $(s, d)$  paths is generated. *Heuristic(.)* then processes all the  $(s, d)$  paths and returns a candidate path to serve the LSP request. If *Heuristic(.)* returns multiple paths, the algorithm selects the one with the fewest number of hops; if there is a tie, the first path is selected. If all the links on the selected candidate path are able to meet the required bandwidth demand, the path is assigned to  $(s, d)$ . The algorithm then subtracts the requested demand from the available bandwidth, see line-9, of each link on the established LSP and each of these links are marked as active permanently. On the contrary, if the selected candidate path is not able to serve the requested demand, *Heuristic(.)* evaluates the remaining paths of  $q$ . If no paths

with sufficient bandwidth is found, it rejects LSP request  $q$  and moves to the next one.

---

**Algorithm 1** Pseudocode for offline heuristics
 

---

**Var:**

$cand\_path$ : a variable that stores a candidate LSP

$links\_used$ : an array that stores used links

$a$ : number of accepted requests

$r$ : number of rejected requests

```

1:  $a = r = 0, links\_used = \emptyset$ 
2: Generate set  $Q$ 
3: for each  $q = \langle s, d, bw \rangle \in Q$  do
4:   Generate  $P_q$ 
5:   while  $P_q \neq \{\}$  do
6:      $cand\_path \leftarrow Heuristic(P_q)$ 
7:     if all links in  $cand\_path$  satisfy  $c_{ij} \geq bw$  then
8:        $LSP \leftarrow cand\_path$ 
9:        $UpdateBandwidth(LSP, bw)$ 
10:       $links\_used \cup \{e_{ij} \mid e_{ij} \in cand\_path\}$ 
11:       $a++$ 
12:    else
13:       $P_q - \{cand\_path\}$ 
14:      if  $P_q = \{\}$  then
15:         $r++$ 
16:      end if
17:    end if
18:  end while
19: end for

```

---

#### 4.1.1 Offline-MO

The goal is to select paths that share the most links. *Offline-MO* compares each  $p_q^k$  of a given request  $q$  with the candidate paths of other requests in  $Q$ . For each  $p_q^k$ , where  $k = 1, 2, \dots, |P_q|$ , the function *Heuristic(.)* finds and stores the number of matching links in a variable  $score \geq 0$  that gives the total number of its links that are in common with paths for other requests.

*Offline-MO* selects  $p_q^k$  that has the maximum  $score$  value. The links within the chosen  $p_q^k$  are then added into  $links\_used$ . If a given  $p_q^k$  has insufficient bandwidth, which depends on the MLU of the different links composing that path, it is removed from  $P_q$ .

An example is presented in Table 2. Here, we only show how *Offline-MO* selects the candidate LSP for request 1; i.e.,  $LSP_1$  ( $\langle R3, R1, 14 \text{ Mb} \rangle$ ). Other LSP requests are processed in the same way. From Table 1, the generated paths in  $P_q$  for  $LSP_1$  are the following:  $p_1^1 = [R3, R1]$ ,  $p_1^2 = [R3, R2, R1]$ ,  $p_1^3 = [R3, R4, R2, R1]$ . Each cell of Table 2 contains the links of candidate paths that belong to other requests; e.g., the second candidate path of  $LSP_2$ ,  $p_2^2$  is  $[R4, R3, R2]$  and its links are  $R4 - R3$  and  $R3 - R2$ . Next to each link is a label that indicates whether it is used by a path belonging to  $LSP_1$ . For example, the path  $[R4, R3, R2]$  belonging to  $LSP_2$  has the link  $R3 - R2$  in common with the second candidate path, i.e.,  $p_1^2$  of  $LSP_1$ , i.e.,  $[R3, R2, R1]$ . This explains the “ $p_1^2$ ” below link  $R3 - R2$ . Links can be shared multiple times as is the case of  $R3 - R4$ , which is shared by  $p_2^2$  and  $p_3^3$ .

From Table 2, we can see the  $score$  for each of the  $P_q$  paths for  $LSP_1$ . For example, for  $p_1^1$ , i.e.,  $[R3, R1]$ , its  $score$  is 1, given that  $p_1^1$  appears one time. The  $score$  for  $[R3, R2, R1]$  is 2, and the  $score$  for  $[R3, R4, R2, R1]$  is 4. This means the *Heuristic(.)* function for  $LSP_1$  returns  $[R3, R4, R2, R1]$  as the path that has the highest overlap and, therefore, is chosen by *Offline-MO*.

#### 4.1.2 Offline-R

Similar to *Offline-MO*, *Offline-R* aims to use paths that have as many common links as possible to other paths and, additionally, gives preference to the ones that require the fewest number of new links to be set up. Note, *new links* are defined as those that are not carrying any traffic, i.e., not in  $links\_used$ . In order to do this, we reuse the  $score$  variable from the *Offline-MO* algorithm and introduce the variable  $Ratio_{off}$  for each of the  $p_q^k$  paths of a given request  $q$ . The said variable is defined as  $score/new\_links\_number$ , where  $new\_links\_number$  stores the number of new links that would have to be established if path  $p_q^k$  is selected. The function *Heuristic(.)* calculates the  $Ratio_{off}$  for each of the  $p_q^k$  paths and selects one with the maximum  $Ratio_{off}$  value. In the special case when  $new\_links\_number$  is equal to 0, i.e., all the links in a given  $p_q^k$  path already exist, the variable is set to 1. This is to avoid division by 0.

Table 3 presents an example with two LSP requests:  $LSP_1$ ,  $\langle R3, R1, 14 \text{ Mb} \rangle$ , and  $LSP_2$ ,  $\langle R4, R2, 41 \text{ Mb} \rangle$ .

**Table 2** *Offline-MO* example.  $LSP_1$  has a request  $\langle R3, R1, 14 \text{ Mb} \rangle$ , and three candidate paths:  $[R3, R1]$ ,  $[R3, R2, R1]$ ,  $[R3, R4, R2, R1]$

$LSP_q$	$p_q^1$ links	$p_q^2$ links	$p_q^3$ links
2	$R4 - R2$	$R4 - R3$	$R3 - R2$
	$p_1^3$		$p_1^2$
			$p_1^1$
3	$R2 - R4$	$R2 - R3$	$R3 - R4$
		$p_1^3$	$p_1^2$
			$p_1^1$

**Table 3** Offline-R example.  $LSP_1$  has a request  $\langle R3, R1, 14 \text{ Mb} \rangle$ , and  $LSP_2$ ,  $\langle R4, R2, 41 \text{ Mb} \rangle$ , and three candidate paths:  $[R3, R1]$ ,  $[R3, R2, R1]$ ,  $[R3, R4, R2, R1]$ , and  $[R4, R2]$ ,  $[R4, R3, R2]$ ,  $[R4, R3, R1, R2]$ , respectively

$LSP_q$	$links\_used$	$k$ paths	score	$new\_links\_number$	$Ratio_{off}$	Selected path $k$
1	{}	$[R3, R1]$	1	1	1	$[R3, R4, R2, R1]$
		$[R3, R2, R1]$	2	2	1	
		$[R3, R4, R2, R1]$	4	3	1.33	
2	$R3 - R4$	$[R4, R2]$	1	0	1	$[R4, R2]$
	$R4 - R2$	$[R4, R3, R2]$	0	2	0	
	$R2 - R1$	$[R4, R3, R1, R2]$	1	3	0.33	

Each of them has three candidate paths  $[R3, R1]$ ,  $[R3, R2, R1]$ ,  $[R3, R4, R2, R1]$ , and  $[R4, R2]$ ,  $[R4, R3, R2]$ ,  $[R4, R3, R1, R2]$ , respectively. The  $score$  variable is calculated similarly to the *Offline-MO* heuristic. For example, consider the first arriving LSP request,  $LSP_1$ , with path  $[R3, R4, R2, R1]$ , and  $score$  value of 4; see Table 2. Given that there is no established link, the variable  $new\_links\_number$  will be set to 3 as links  $R3-R4$ ,  $R4-R2$ , and  $R2-R1$  need to be set up. Therefore, the value of  $Ratio_{off}$  is  $4/3 = 1.33$ . The other two paths of  $LSP_1$ , i.e.,  $[R3, R1]$  and  $[R3, R2, R1]$ , will also need new links to be established, and their  $Ratio_{off}$  value is  $1/1 = 1$  and  $2/2 = 1$ .  $Heuristic(\cdot)$  will then return path  $[R3, R4, R2, R1]$  as the candidate path given that its  $Ratio_{off}$  is the maximum among these three paths. For  $LSP_2$  and subsequent LSP requests, paths needing more new links to be set up will have less chance of being selected as LSP candidates.

## 4.2 Online approaches

Algorithm 2 presents the pseudocode for our online heuristics. Note that this pseudocode is similar to the pseudocode presented in Algorithm 1. The difference is that by definition, online approaches only have knowledge of the current and past LSP requests.

### 4.2.1 Online-MO

Following a similar criterion to its offline counterpart, *Online-MO* selects  $p_q^k$  with links that overlap the most with existing links.  $Heuristics(\cdot)$  calculates for each  $p_q^k$  the number of links in common with already established links and stores this in the  $num\_used\_link$  variable. The  $p_q^k$  path with the maximum number of links in common is selected as the candidate LSP. The variable  $num\_used\_link$  indicates path  $p_q^k$  contains at least one link that is already established and, therefore, can be reused.

Table 4 shows an example for the first two arriving LSP requests. For  $LSP_1$ ,  $num\_used\_link$  is 0 initially for paths  $[R3, R1]$ ,  $[R3, R2, R1]$ , and  $[R3, R4, R2, R1]$ . Here,  $Heuristic(\cdot)$  will break the tie by returning the shortest path  $[R3, R1]$  as the candidate path. This is not the case for  $LSP_2$  given that  $Heuristic(\cdot)$  will consider that link  $R3-R1$  has been setup. The paths for  $LSP_2$  are

## Algorithm 2 Pseudocode for online heuristics

**Var:**

$cand\_path$ : a variable that stores the candidate LSP

$links\_used$ : a table that stores the links used

$a$ : number of accepted requests

$r$ : number of rejected requests

```

1: Request  $q = \langle s, d, bw \rangle$  arrives
2: Generate  $P_q$ 
3: while  $P_q \neq \{\}$  do
4:    $cand\_path \leftarrow Heuristic(P_q)$ 
5:   if all links in  $cand\_path$  satisfy  $c_{ij} \geq bw$  then
6:      $LSP \leftarrow cand\_path$ 
7:      $UpdateBandwidth(LSP, bw)$ 
8:      $links\_used \cup \{e_{ij} \mid e_{ij} \in cand\_path\}$ 
9:      $a++$ 
10:  else
11:     $P_q - \{cand\_path\}$ 
12:    if  $P_q = \{\}$  then
13:       $r++$ 
14:    end if
15:  end if
16: end while

```

$[R4, R2]$ ,  $[R4, R3, R2]$ , and  $[R4, R3, R1, R2]$ . Their corresponding  $num\_used\_link$  value is 0, 0, and 1, respectively, given that path  $[R4, R3, R1, R2]$  is the only one that can reuse link  $R3-R1$ . Therefore,  $Heuristic(\cdot)$  will return as candidate path  $[R4, R3, R1, R2]$ .

**Table 4** Online-MO example.  $LSP_1$   $\langle R3, R1, 14 \text{ Mb} \rangle$  and  $LSP_2$   $\langle R4, R2, 41 \text{ Mb} \rangle$ 

$LSP_q$	$links\_used$	$k$ paths	$num\_used\_link$	Selected path $k$
1	{}	$[R3, R1]$	0	$[R3, R1]$
		$[R3, R2, R1]$	0	
		$[R3, R4, R2, R1]$	0	
2	$R3 - R1$	$[R4, R2]$	0	$[R4, R3, R1, R2]$
		$[R4, R3, R2]$	0	
		$[R4, R3, R1, R2]$	1	

**Table 5** *Online-R* example.  $LSP_1 \langle R3, R1, 14 \text{ Mb} \rangle$  and  $LSP_2 \langle R4, R2, 41 \text{ Mb} \rangle$

$LSP_q$	$links\_used$	$k$ paths	$num\_used\_link$	$new\_links\_number$	$Ratio_{on}$	Selected path $k$
1	{}	[R3, R1]	0	1	0	[R3, R1]
		[R3, R2, R1]	0	2	0	
		[R3, R4, R2, R1]	0	3	0	
2	R3 – R1	[R4, R2]	0	1	0	[R4, R3, R1, R2]
		[R4, R3, R2]	0	2	0	
		[R4, R3, R1, R2]	1	2	0.5	

**4.2.2 Online-R**

This heuristic is similar to its offline counterpart; i.e., *Offline-R*. The objective here is to reduce energy consumption by utilizing established links and, additionally, favoring paths that require the fewest new links.

Note that this approach is similar to that of [27]. Specifically, for the routing of a given  $(s, d)$  pair demand, the technique in [27] uses an existing shortest backup path. It aims to minimize the establishment of new links. In contrast, *Online-R* does not only consider backup paths of a given  $(s, d)$  pair paths but considers the shortest paths used to route demands for other  $(s, d)$  pairs. This helps reduce the need to establish new links.

We reuse the term  $num\_used\_links$  from the *Online-MO* algorithm and  $new\_links\_number$  from *Offline-R* and introduce the term  $Ratio_{on}$  as the ratio  $num\_used\_links/new\_links\_number$ . For each  $p_q^k$  paths of the current request  $q$ ,  $Heuristic(.)$  calculates its  $Ratio_{on}$  and then selects as the candidate LSP  $p_q^k$  whose  $Ratio_{on}$  is maximum. In the special case when  $new\_links\_number$  is equal to zero, i.e., all the links in a given  $p_q^k$  path have already been established, the variable is set to 1. This is to avoid division by 0.

Table 5 describes how *Online-R* calculates the candidate LSP for the first two LSP requests. Notice that paths that

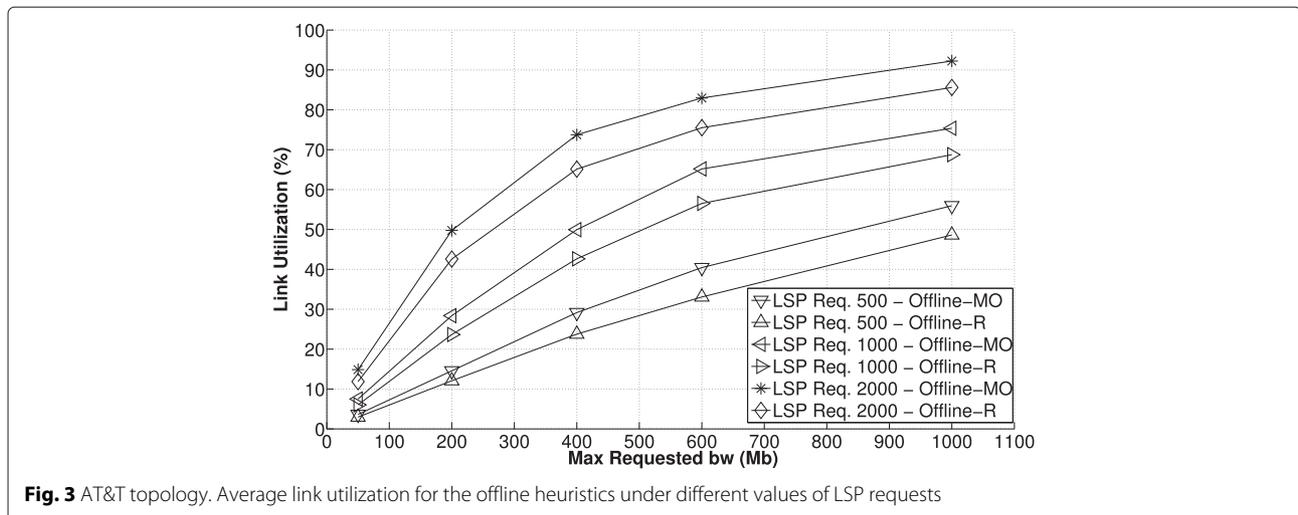
require more new links to be setup have a lower probability of being selected as a candidate LSP, while those that need fewer new links are preferred.

As an example, any of the paths in  $LSP_1$ , i.e., [R3, R1], [R3, R2, R1], and [R3, R4, R2, R1], will require all their links to be setup. Therefore, their  $num\_used\_links$  value will be zero, and their ratio will also be zero.  $Heuristic(.)$  will break the tie by selecting the shortest path [R3, R1].

We now turn our attention to  $LSP_2$ . Its paths, i.e., [R4, R2], [R4, R3, R2], and [R4, R3, R1, R2], will have a  $num\_used\_links$  value of 0, 0, and 1, respectively. The value of  $new\_links\_number$  for each of candidate path of  $LSP_2$  can be found by counting the links that are not included in the *Links used* column. Specifically, the corresponding  $new\_links\_number$  value for [R4, R2], [R4, R3, R2], and [R4, R3, R1, R2] is 1, 2, and 2, respectively. Given the value of  $num\_used\_links$  and  $new\_links\_number$ ,  $Ratio_{on}$  can be calculated and  $Heuristic(.)$  returns the candidate path with highest value. In this case, path [R4, R3, R1, R2] is selected.

**4.2.3 Online-MinH**

This heuristic, which is also reported in [28], chooses as a candidate LSP, the  $p_q^k$  path of the current request  $q$  with



**Fig. 3** AT&T topology. Average link utilization for the offline heuristics under different values of LSP requests

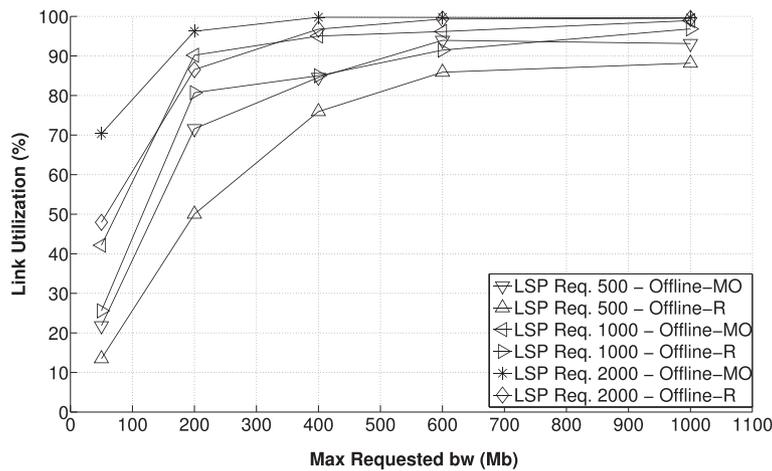


Fig. 4 Abilene topology. Average link utilization for the offline heuristics under different values of LSP requests

the minimum number of hops. We skip its example due to its simplicity.

#### 4.2.4 Online-R-LSP

The *Heuristic(.)* for *Online-R-LSP* randomly selects one of the  $p_q^k$  paths in the set  $P_q$  for the current request  $q$ . Note that random path selection is essentially similar to Equal-Cost Multipath (ECMP), as used by CSPF [29].

### 5 Evaluation

The performance of the aforementioned heuristics is evaluated using two popular topologies: Abilene and AT&T North America [59, 60]. The Abilene network consists of 11 nodes and 28 directional links, whereas the AT&T network consists of 25 nodes and 112 directional links.

We conducted our simulations in MATLAB [61]. The three components of a LSP request  $\langle s, d, bw \rangle$  are generated randomly as follows: i)  $s$  and  $d$  are sets to an integer from the range  $[1, |V|]$ , where  $s \neq d$ , ii)  $bw$  is a value in  $[1, BW_{Max}]$ . LSP requests are generated in advance in

both online and offline scenarios. We assume that when all links are active, all these LSP requests can be admitted.

Algorithm 3 describes the procedure used for all simulations. Please note steps 4 and 7. These steps show the calculations performed by *Heuristic(.)* for a given set of LSP requests  $Q$ . In particular, for all requests  $q$  in  $Q$ , *Heuristic(.)* needs to explore  $|P_q|$  shortest paths; each of them with a maximum length of  $|V|$  hops. Therefore, our algorithms have a running time complexity of  $O(|P_q||Q||V|)$ .

---

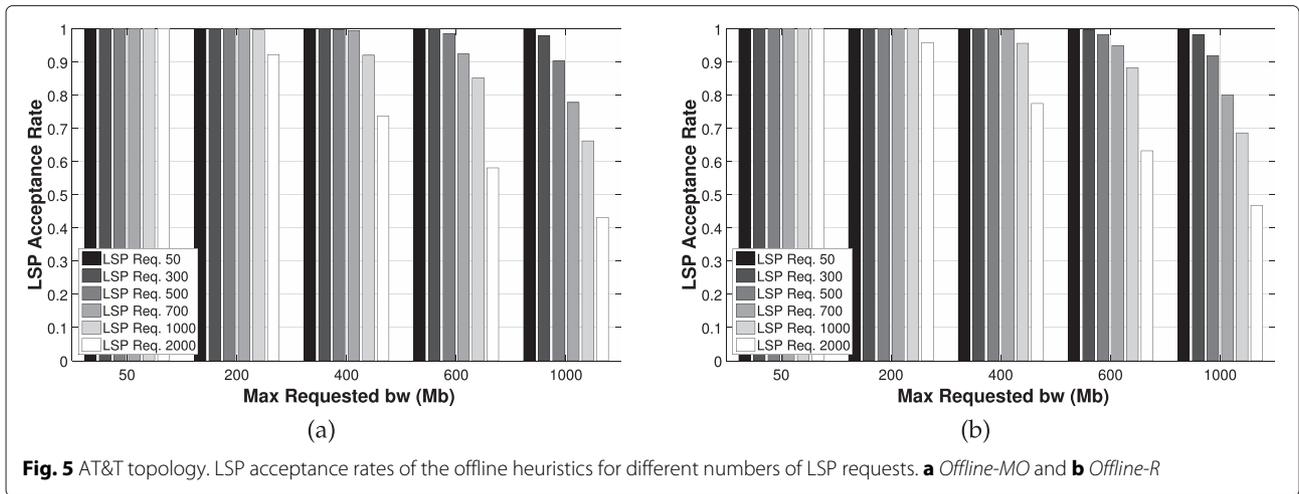
#### Algorithm 3 Pseudocode for different simulation.

---

- 1: Generate  $Q$
  - 2: **if** heuristic is offline **then**
  - 3:   Call offline *Heuristic(.)* with  $Q$  as its input.
  - 4:   Assign or reject the request  $q$  according to offline *Heuristic(.)*, and compute the accepted/rejected LSP requests.
  - 5: **else**
  - 6:   Call online *Heuristic(.)* with current request  $q$  as its input
  - 7:   Assign or reject the request  $q$  according to online *Heuristic(.)*, and compute the accepted/rejected LSP requests.
  - 8: **end if**
  - 9: **for each**  $link(i, j)$  **do**
  - 10:   Calculate  $u_{ij}$
  - 11: **end for**
  - 12: Calculate LSP acceptance rate
  - 13: Calculate the number of active links in the final topology
  - 14: Compute  $\rho$
- 

Table 6 Overall acceptance rate of offline heuristics over AT&T and Abilene

Max req. bw (Mb)	Offline			
	AT&T		Abilene	
	Offline-R	Offline-MO	Offline-R	Offline-MO
50	1.0	1.0	1.0	0.99
200	0.99	0.99	0.88	0.81
400	0.96	0.94	0.71	0.62
600	0.91	0.89	0.59	0.52
1000	0.82	0.79	0.47	0.42
Average	0.94	0.92	0.73	0.67



In order to measure the goodness of a solution, we define a new metric  $\rho = PSL/(100-LAR)$ . Recall that PSL is the percentage of shutdown links and LAR is the LSP acceptance rate. Consider two green LSP methods:  $LSP_A$  and  $LSP_B$ . Assume both can shut down the same number of links. For instance,  $PSL_A = PSL_B = 40\%$ . However, let us assume they have an LSP acceptance rate of  $LAR_A = 90\%$  and  $LAR_B = 80\%$ , respectively. Therefore, we have  $\rho_A = 4$  and  $\rho_B = 2$ . Given that  $\rho_A > \rho_B$ , we conclude that  $LSP_A$  is better than  $LSP_B$ . Consider a second example. Let us assume that  $PSL_A = 30\%$  and  $PSL_B = 40\%$ , and both have the same LAR, say  $LAR_A = LAR_B = 70\%$ . Therefore,  $\rho_A = 1$  and  $\rho_B = 1.3$ , and  $\rho_B > \rho_A$ . In this case,  $LSP_B$  is better because it is able to shut down a larger number of links while keeping the same LSP acceptance rate. Please note that when a green approach attains  $LAR = 100\%$ , its  $\rho$  will go to infinity. In this case, we set  $\rho$  to  $PSL$ .

We conducted 30 simulation runs for each of the heuristics discussed in Section 4 using the following number of arriving LSP requests ( $|Q|$ ): 50, 300, 500, 700, 1000, and

2000. In order to simulate different network loads, for each  $|Q|$  value, we set  $BW_{Max}$  to 50, 200, 400, 600, and 1000 Mb/s. Finally, we compute  $\rho$  for each of the evaluated approaches.

In the following sections, note that *low* network load refers to scenarios with no more than 300 LSP requests and their max requested bandwidth is less than or equal to 200 Mb. Conversely, we use the term *high* network load for scenarios where the number of LSP requests is at least 1000 and their max requested bandwidth is greater than or equal to 600 Mb.

### 5.1 Offline approaches

Figures 3 and 4 present the average link utilization of the two proposed offline heuristics for the AT&T and Abilene networks, respectively. As expected, under low network load, link utilization is similar for both approaches as few links are used and similar LSPs are selected. For both topologies, *Offline-R* presents the best performance with

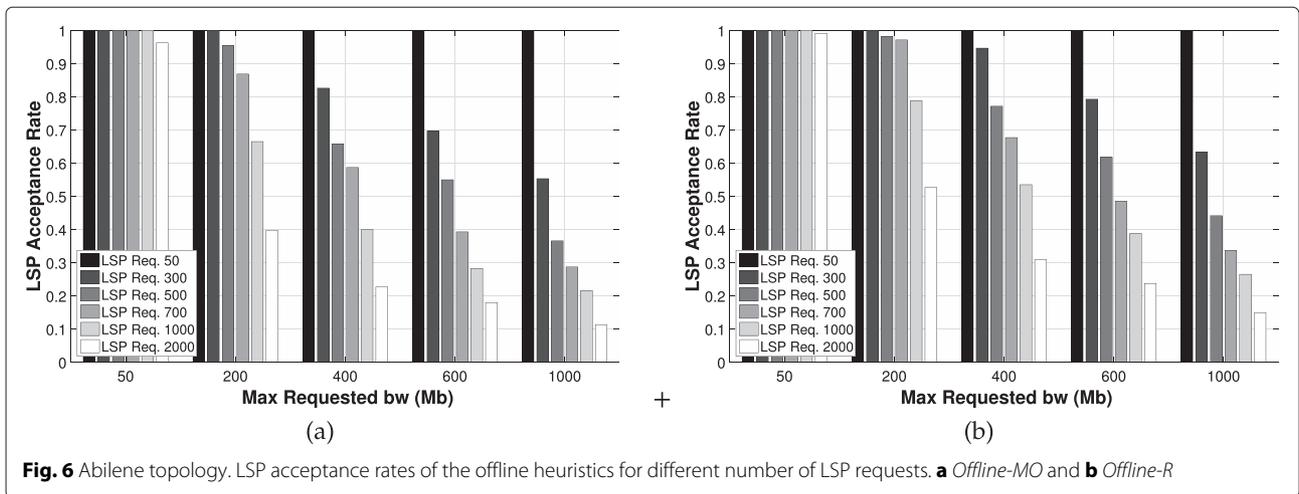


Fig. 6 Abilene topology. LSP acceptance rates of the offline heuristics for different number of LSP requests. a Offline-MO and b Offline-R

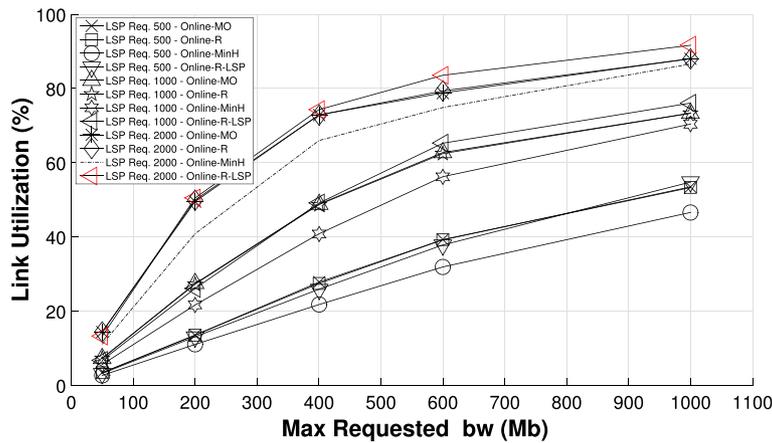


Fig. 7 AT&T topology. Average link utilization for the online heuristics under different values of LSP requests

an average link utilization of 28.18 and 59.4 % for AT&T and Abilene, respectively. This is due to its ability to reuse established links, which allows it to obtain the lowest overall link utilization, whereas *Offline-MO* presents an average link utilization of 32.4 and 68.6 % for AT&T and Abilene, respectively. Recall that *Offline-R* and *Offline-MO* aim to use paths that share links as much as possible with other paths. However, *Offline-R* has the advantage of preferring paths that require the least number of new links to be established. Hence, it has the best overall performance. However, both approaches show a similar rate of increase in their average link utilization at higher network load.

For AT&T, if the network load is low with  $|Q| = 50$ , *Offline-R* has an average link utilization of 2.9 % against 3.23 % for *Offline-MO*. In high network load scenarios, i.e.,  $|Q| = 2000$ , the average link utilization of *Offline-R* reaches 56.13 % versus 62.71 % for *Offline-MO*; this indicates an increase of 1935.5 and 1941.4 %, respectively.

These results are also consistent for Abilene. Under low network load, i.e.,  $|Q| = 50$ , *Offline-R* has an average link utilization of 12.64 % and *Offline-MO* 18.18 %; when the network load is increased to  $|Q| = 2000$ , the average link utilization rises to 86.01 and 93.15 %, respectively, which means a rate of increase of 680.45 % for *Offline-R* and 512 % for *Offline-MO*. The average link utilization of Abilene is consistently 2.1 times that of AT&T under the same network load. This is because Abilene has fewer links, or smaller network capacity. This difference in link utilization has a direct impact on the final number of active links.

Table 6 presents the average LSP acceptance rate for AT&T and Abilene topologies. Figures 5 and 6 depict the data in Table 6 for AT&T and Abilene, respectively. For AT&T, both approaches show good performance. In particular, *Offline-R* exhibits a slightly better overall LSP acceptance rate of 93 % as compared to 92 % for *Offline-MO*.

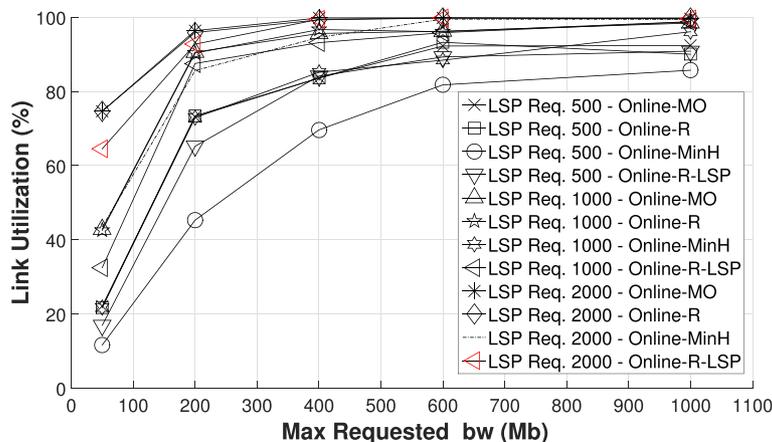


Fig. 8 Abilene topology. Average link utilization for the online heuristics under different values of LSP requests

**Table 7** Online heuristics that exhibit the largest LSP acceptance rates for AT&T and Abilene

Max req. bw (Mb)	Online			
	AT&T		Abilene	
	<i>Online-MinH</i>	<i>Online-R-LSP</i>	<i>Online-MinH</i>	<i>Online-R-LSP</i>
50	1.0	1.0	1.0	1.0
200	0.99	1.0	0.89	0.86
400	0.96	0.96	0.74	0.67
600	0.92	0.92	0.63	0.55
1000	0.83	0.82	0.49	0.45
Average	0.94	0.94	0.75	0.71

Figure 5 shows the LSP acceptance rate for this scenario is above 40 %. This is in spite of the average link utilization being above 80 % as observed in Fig. 3.

With respect to Abilene, Fig. 6 shows that both approaches present a similar performance with an average LSP acceptance rate of 73 % for *Offline-R* and 67 % for *Offline-MO*. However, given that the network utilization of Abilene increases more rapidly than AT&T, the observed LSP acceptance rate also decreases significantly; as an example, consider the case when the number of LSP requests is 2000 and the max requested bandwidth is 400; for AT&T, the LSP acceptance rate is above 70 % for both approaches, whereas for Abilene, the acceptance rate is below 40 %.

## 5.2 Online approaches

Figures 7 and 8 present the average link utilization for the different online heuristics. In the case of AT&T, the lowest average link utilization is observed for *Online-MinH*, at 27.32 %, and the second lowest average link utilization of 31.19 % belongs to *Online-R*, which utilizes established links and prefers paths that require the fewest number of

new links to be set up. The overall link utilization is low for both heuristics. This is because most LSPs are routed over fewer links and leaving many links little to no load.

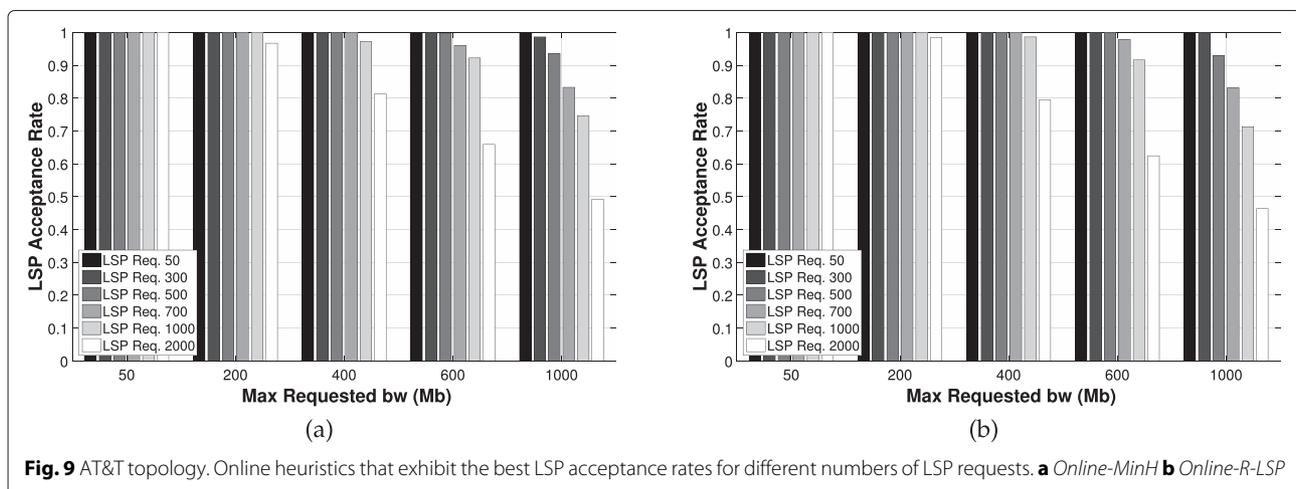
In the case of Abilene, *Online-MinH* also produces the lowest average link utilization at 56.48 %. Surprisingly, the second best performer at 65.2 % is *Online-R-LSP* that selects LSPs randomly. However, the utilization of *Online-R-LSP* is very close to that of other approaches. As expected, when the network load increases, link utilization also increases. In particular, *Online-MinH* shows an increase of 53.44 % and 74.68 % for AT&T and Abilene respectively when going from the lowest to the highest possible network loads.

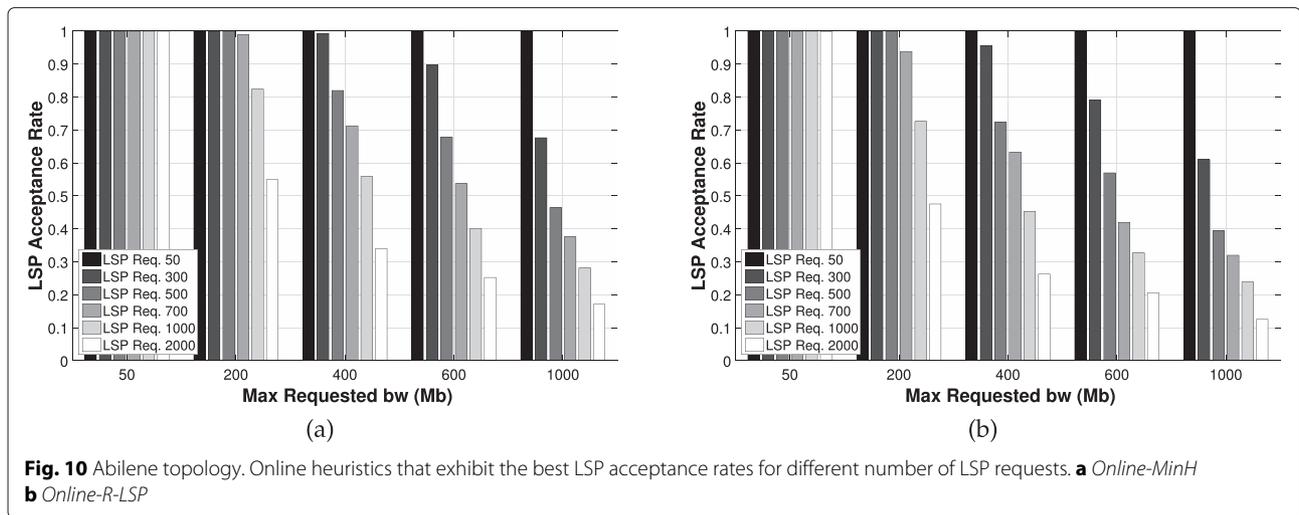
Table 7 shows the average LSP acceptance rate for the online heuristics that exhibit the highest LSP acceptance rates for AT&T and Abilene. Figures 9 and 10 are plots of Table 7.

*Online-MinH* and *Online-R-LSP* exhibit the best performance for both topologies. Overall, *Online-MinH* has a slightly better performance than *Online-R-LSP*. Specifically, for AT&T, *Online-MinH* and *Online-R-LSP* present the same overall LSP acceptance rate of 94 %. For Abilene, *Online-MinH*, shows an average LSP acceptance rate of 75 % against 71 % for *Online-R-LSP*.

These LSP acceptance rates are due to *Online-MinH* attaining the lowest average link utilization for both topologies, see Figs. 7 and 8.

Note that the total average LSP acceptance rate for the online approaches, 83.5 %, is larger than the total average LSP acceptance rate for the offline approaches, 81.5 %. These total average LSP acceptance rates are obtained by computing the mean of the average values presented in Tables 6 and 7, respectively. The main goal of offline approaches when establishing LSPs is to minimize the overall energy consumption of the network even if this implies a decrease in LSP acceptance rates. On the other hand, *Online-MinH* and *Online-R-LSP* do not consider





energy savings as the main factor when establishing LSPs and their main objective is to accept as many future LSP requests as possible. Consequently, they have higher LSP acceptance rates. This trade-off between energy savings and LSP acceptance rates will be discussed in more detail in Section 5.3 when we compare the tested heuristics according to their  $\rho$  ratio.

### 5.3 Discussion

Tables 8 and 9 show a comparison between the percentage of shut down links achieved by online approaches and the overall percentage of shut down links for the best offline approach; namely, *Offline-R*. *Offline-R* is able to shut down 22.1 and 8.6 % of the links in AT&T and Abilene, respectively. We remark that *Offline-MO* exhibits a much lower percentage of shut down links than *Offline-R*; we thus omit the results for *Offline-MO*. *Offline-MO* shuts 9.9 % of the links for AT&T and less than 1 % for Abilene. The tables also show the LSP acceptance rates for *Offline-R* and online approaches. The  $\rho$  ratio for each heuristic is also presented. The better performance exhibited by *Offline-R* is the result of its LSP selection policy that requires fewer new links to be established, which has a direct impact on the final number of active links.

It is interesting to see that *Online-MinH* and *Online-R-LSP* are among the approaches with the worst performance in regard to the overall percentage of shut down links, with 17 and 5.7 %, for AT&T, and 2.13 and 0.12 %, for Abilene, respectively. At the same time, these two approaches have the highest LSP acceptance rate. *Online-MinH* has an overall LSP acceptance rate of 94 and 75 % for AT&T and Abilene, respectively. The corresponding values for *Online-R-LSP* are 94 and 71 % for AT&T and Abilene, respectively. On the other hand, *Online-MO* and *Online-R* show the lowest LSP acceptance rate; both recorded a percentage of 92 % for AT&T and 67 % for Abilene, respectively. However, these two approaches are the ones that were able to shut down the largest percentage of links. For AT&T, the percentage of shut down links when using these approaches is 21.6 and 19.3 %, respectively. When tested over Abilene, *Online-R* exhibits a slightly better performance than *Online-MO*; i.e., 2.9 versus 2.1 %, respectively. As expected, there is a clear trade-off between LSP acceptance rates and the number of active links. The larger the LSP acceptance rate, the fewer the number of links a green technique is able to shut down. The good performance of *Online-MO* and *Online-R* is due to their low overall link utilization; see Figs. 7 and 8.

**Table 8** Comparison of the performance of online approaches and *Offline-R* according to their  $\rho$  ratio for AT&T

Heuristic	Percentage of shut down links (%)	Percentage of links shut down by <i>Offline-R</i>	Overall LSP acceptance rate (%)	$\rho$
<i>Offline-R</i>	22.1	100	94	3.7
<i>Online-MO</i>	21.6	97.7	92	2.7
<i>Online-R</i>	19.3	87.3	92	2.4
<i>Online-MinH</i>	17.0	76.9	94	2.8
<i>Online-R-LSP</i>	5.7	12.2	94	0.95

**Table 9** Comparison of the performance of online approaches and *Offline-R* according to their  $\rho$  ratio for Abilene

Heuristic	Percentage of shut down links (%)	Percentage of links shut down by <i>Offline-R</i>	Overall LSP acceptance rate (%)	$\rho$
<i>Offline-R</i>	8.6	100	73	0.31
<i>Online-R</i>	2.9	33.4	67	0.08
<i>Online-MO</i>	2.14	24.9	67	0.06
<i>Online-MinH</i>	2.13	24.8	75	0.09
<i>Online-R-LSP</i>	0.12	1.4	71	0.004

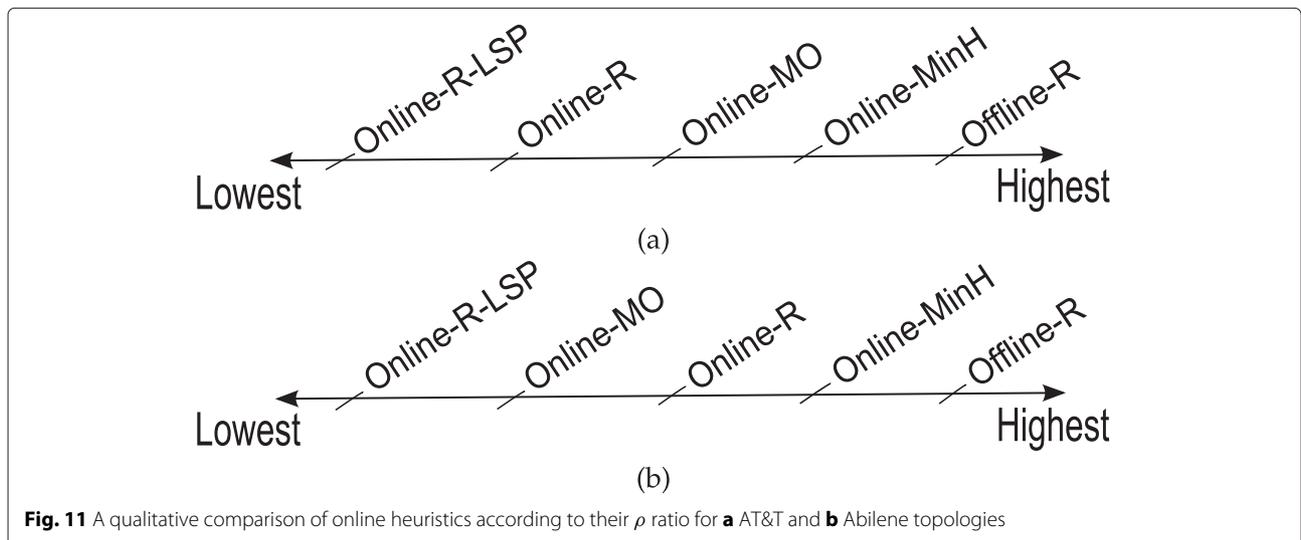
Table 8 also shows that for AT&T, the percentage of shut down links for the best online approach, *Online-MO*, is around 97.7 % of the percentage of shut down links observed for *Offline-R*. *Online-MO* selects paths that require the fewest number of new links, which decreases the overall percentage of active links. On the other hand, *Online-R-LSP* randomly selects paths without considering energy consumption. This results in *Online-R-LSP* exhibiting the smallest percentage of shut down links among the studied approaches, with only 12.2 % of the recorded percentage of *Offline-R*. For Abilene, Table 9 indicates that *Online-R* and *Online-R-LSP* exhibit the best and worst performance, respectively. Specifically, *Online-R* is able to shut down 24.9 % of the links shut down by *Offline-R*, whereas, *Online-R-LSP* only shuts down 1.4 % of the links shut down by *Offline-R*.

Finally, we study  $\rho$ ; see Fig. 11. The figure qualitatively compares the performance of all the studied approaches for the AT&T and Abilene topologies. *Online-MinH* is the best online approach with a  $\rho$  ratio of 3.7. This is an interesting result given that *Online-MO* and *Online-R* are the approaches that present the largest energy savings. Note that *Online-MinH* exhibits better LSP acceptance rates than *Online-MO* and *Online-R*. However, its percentages of shut down links are slightly smaller than the

percentages exhibited by *Online-MO* and *Online-R*. This means that in this case, it is better to sacrifice some energy in exchange for better LSP acceptance rates. Observe that *Online-MO* presents the third and fourth best performance for AT&T and Abilene, respectively. On the other hand, *Online-R* presents the fourth and third best performance for AT&T and Abilene, respectively. In this particular case, both approaches present the same LSP acceptance rate regardless of the topology. However, the approach with the higher percentage of shut down links is the one that exhibits the best overall performance.

### 6 Conclusions

In this paper, we have analyzed the problem of reducing the energy consumption of an MPLS network using online and offline path establishment methods. We believe this to be the first extensive work that studies green LSP establishment solutions. We study six heuristics over the same topologies. Notably, we compare online and offline heuristics in terms of energy savings and LSP acceptance rates. On the Abilene and AT&T topologies, results indicate that during off-peak periods, LSP acceptance rates above 90 % are possible with 20 % of links shut down to conserve energy.



**Fig. 11** A qualitative comparison of online heuristics according to their  $\rho$  ratio for **a** AT&T and **b** Abilene topologies

As a future work, we plan to extend the work in [62] to maximize the number of accepted connections and channel assignment in a green network. We remark that although we assume a MPLS network, the online and offline heuristics may be applicable to other networks. This is because a LSP can be interpreted as a path or connection from a source node. For example, establishing paths in wireless sensor networks in an online or offline manner [63] with the goal of minimizing the number of involved sensor nodes so that other nodes can conserve their energy. Another possible direction is to consider multicast; e.g., [64]. The problem then is to construct a tree using the minimal number of nodes that supports all arriving demands. Lastly, implications on security will have to be considered; see [65, 66].

#### Competing interests

The authors declare that they have no competing interests.

#### Authors' contributions

The first author designed all heuristics, implemented all said heuristics, conducted experiments, and drafted the paper. All other authors contributed in terms of research direction, e.g., problem definition and formulation, designing experiments and in writing/editing the paper. All authors read and approved the final manuscript.

#### Acknowledgements

Author Alejandro Ruiz-Rivera acknowledges and thanks the support of Catholic Education network (CENet) and its staff.

#### Author details

<sup>1</sup>School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, Australia. <sup>2</sup>Department of Computing, Curtin University, Perth, Australia.

Received: 4 May 2015 Accepted: 21 June 2015

Published online: 16 July 2015

#### References

- Enabling the low carbon economy in the information age (2008). Available: [www.smart2020.org/publications/](http://www.smart2020.org/publications/)
- F Ramos, RJ Gibbens, F Song, P Rodriguez, J Crowcroft, IH White, in *ACM SIGCOMM Workshop on Green Networking*. Reducing Energy Consumption in IPTV Networks by Selective Pre-joining of Channels (New Delhi, India, 2010)
- A Bacioccola, C Cicconetti, G Stea, in *Proc. of Intl. Conf. on Performance Evaluation Methodologies and Tools*. User-level Performance Evaluation of VoIP Using ns-2 (Nantes, France, 2007)
- D Singh, G Tripathi, AJ Jara, in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. A survey of internet-of-things: future vision, architecture, challenges and services (Seoul, Korea (South), 2014), pp. 287–292
- BP Rimal, E Choi, I Lumb, in *Fifth International Joint Conference on INC, IMS and IDC, 2009. NCM '09*. A taxonomy and survey of cloud computing systems (Seoul, Korea, 2009), pp. 44–51
- The Zettabyte Era: Trends and Analysis (2014). [http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI\\_Hyperconnectivity\\_WP.pdf](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.pdf)
- GP Intl, Cloud computing and its contribution to climate change. Available: <http://goo.gl/nGveJm> (2010). [www.greenpeace.org/international/en/publications/reports/](http://www.greenpeace.org/international/en/publications/reports/)
- G Ananthanarayanan, RH Katz, in *Proc. of Power Aware Computing and Systems*. Greening the Switch (San Diego, CA, 2008)
- M Gupta, S Singh, in *ACM SIGCOMM*. Greening of the Internet (Karlsruhe, Germany, 2003)
- S Nedeveschi, L Popa, G Iannaccone, S Ratnasamy, D Wetherall, in *ACM/USENIX NSDI*. Reducing Network Energy Consumption Via Sleeping and Rate-adaptation (San Francisco, CA, 2008)
- C Gunaratne, K Christensen, B Nordman, S Suen, Reducing the energy consumption of Ethernet with adaptive link rate (ALR). *IEEE Trans. Comput.* **57**(4), 448–461 (2007)
- M Zhang, C Yi, B Liu, B Zhang, in *IEEE ICNP*. GreenTE: Power-Aware Traffic Engineering (Kyoto, Japan, 2010)
- AP Bianzino, L Chiaraviglio, M Mellia, in *IEEE GreenCom*. GRiDA: A green distributed algorithm for backbone networks, (2011). doi:10.1109/GreenCom.2011.6082517
- A Cianfrani, V Eramo, M Listanti, M Marazza, E Vittorini, in *IEEE INFOCOM*. An Energy Saving Routing Algorithm for a Green OSPF Protocol (San Diego, CA, 2010)
- G Athanasiou, K Tsagkaris, P Vlacheas, D Karvounas, P Demestichas, Multi-objective traffic engineering for future networks. *IEEE Commun. Lett.* **16**(1), 101–103 (2012)
- G Lin, S Soh, KW Chin, M Lazerescu, Efficient heuristics for energy-aware routing in networks with bundled links. *Elsevier Comput. Netw.* **57**(8), 1774–1788 (2013)
- L Wang, F Zhang, AV Vasilakos, C Hou, Z Liu, Joint virtual machine assignment and traffic engineering for green data center networks. *ACM SIGMETRICS Perform. Eval. Rev.* **41**(3), 107–112 (2013)
- L Wang, F Zhang, JA Aroca, AV Vasilakos, K Zheng, C Hou, D Li, Z Liu, GreenDCN: a general framework for achieving energy efficiency in data center networks. *IEEE J. Selected Areas Commun.* **32**(1) (2015). doi:10.1109/JSAC.2014.140102
- F Xu, F Liu, H Jin, AV Vasilakos, Managing performance overhead of virtual machines in cloud computing: a survey, state of the art and future directions. *Proc. IEEE*. **102**(1), 11–31 (2014)
- K Chen, C Hu, X Zhang, K Zheng, Y Chen, AV Vasilakos, Survey on routing in data centers: insights and future directions. *IEEE Netw.* **25**(4), 6–10 (2011)
- XWAV Vasilakos, M Chen, Y Liu, TK Kwon, A survey of green mobile networks: opportunities and challenges. *Springer Mobile Netw. Appl.* **17**(1), 4–20 (2012)
- Z Sheng, S Yang, Y Yu, AV Vasilakos, J McCann, K Leung, A survey of the IETF protocol suite for the internet of things: standards, challenges and opportunities. *IEEE Wireless Commun.* **20**(6), 91–98 (2013)
- J Liu, J Wan, Q Wang, P Deng, K Zhou, Y Qiao, A survey on position-based routing for vehicular ad hoc networks. *Telecommun. Syst.* 36–45 (2015). doi:10.1007/s11235-015-9979-7
- Y Zeng, K Xiang, D Li, AV Vasilakos, Directional routing and scheduling for green vehicular delay tolerant networks. *Springer Wireless Netw.* **19**(2), 161–173 (2013)
- L Xiang, J Luo, A Vasilakos, in *IEEE SECON*. Compressed data aggregation for energy efficient wireless sensor networks (Salt Lake City, USA, 2011), pp. 46–54
- PBF Duarte, ZM Fadlullah, AV Vasilakos, N Kato, On the partially overlapped channel assignment on wireless mesh network backbone: A game theoretic approach. *IEEE JSAC*. **30**(1), 119–127 (2012)
- F Francois, N Wang, K Moessner, S Georgoulas, RdO Schmidt, Leveraging MPLS backup paths for distributed energy-aware traffic engineering. *IEEE Trans. Netw. Serv. Manag.* **11**(2), 235–249 (2014)
- S Avallone, G Ventre, Energy efficient online routing of flows with additive constraints. *Comput. Netw.* **56**(10), 2368–2382 (2012)
- ED Osborne, A Simha, *Traffic Engineering with MPLS*. (Cisco Press, 2002). ISBN-13: 978-1-58705-031-2
- B Nordman, Networks, Energy, and Energy Efficiency. Cisco Green Research Symposium. Available: (Berkeley, Calif. 2008). <http://research.cisco.com/publications.html>
- B Addis, A Capone, G Carello, LG Gianoli, B Sanso, Energy management through optimized routing and device powering for greener communication networks. *IEEE/ACM Trans. Netw.* **22**(1), 313–325 (2014)
- B Addis, A Capone, G Carello, LG Gianoli, B Sanso, in *IEEE International Conference on Communications (ICC) 2012*. Energy-aware multiperiod traffic engineering with flow-based routing (Ottawa, Canada, 2012), pp. 5957–5961
- A Coiro, F Iervini, M Listanti, in *IEEE ICCCN*. Distributed and adaptive interface switch off for internet energy saving (Maui, Hawaii, 2011)
- A Coiro, M Listanti, A Valenti, F Matera, Energy-aware traffic engineering: A routing-based distributed solution for connection-oriented {IP} networks. *Comput. Netw.* **57**(9), 2004–2020 (2013)
- OSPF version 2. RFC 2328. (1998). <http://www.ietf.org/rfc/rfc2328.txt>

36. A Cianfrani, V Eramo, M Listanti, M Polverini, AV Vasilakos, An OSPF-integrated routing strategy for QoS-aware energy saving in IP backbone networks. *IEEE Trans. Netw. Serv. Manag.* **9**(3), 254–264 (2012)
37. L Wang, F Zhang, K Zheng, AV Vasilakos, S Ren, Z Liu, in *IEEE ICDCS*. Energy-efficient flow scheduling and routing with hard deadlines in data center networks (Madrid, Spain, 2014)
38. L Chiaraviglio, M Mellia, F Neri, Minimizing ISP network energy cost: Formulation and solutions. *IEEE/ACM Trans. Netw.* **20**(99), 463–476 (2011). doi:10.1109/TNET.2011.2161487
39. N Wang, C Michael, KH Ho, Disruption-Free green traffic engineering with NotVia Fast Reroute. *IEEE Commun. Lett.* **15**(10), 1123–1125 (2011). doi:10.1109/LCOMM.2011.082611.110769
40. M Shand, S Bryant, S Previdi, IP fast reroute using Not-Via addresses (2011). <http://tools.ietf.org/html/draft-ietf-rtgwg-ipfrr-notvia-addresses-07.txt>
41. G Lin, S Soh, KW Chin, M Lazerescu, Power aware routing in networks with QoS constraints. *Trans. Emerg. Telecommun. Technol.* (2015). doi:10.1109/ICCCNT.2013.6726707
42. K-H Ho, C-C Cheung, in *6th International Conference on Networked Computing (INC)*. Green distributed routing protocol for sleep coordination in wired core networks (yeongju, Korea (South), 2010)
43. H Yonezu, S Gao, S Shimizu, D Ishii, S Okamoto, E Oki, N Yamanaka, in *15th OptoElectronics and Communications Conference (OECC)*. Network Power Saving Topology Calculation Method by Powering Off Links Considering QoS (IEEE, Sapporo, Japan, 2010)
44. N Vasic, D Kostic, in *Proceedings of 1st International Conference on Energy-Efficient Computing and Networking*. Energy-aware Traffic Engineering (Passau, Germany, 2010)
45. F Cuomo, A Abbagnale, A Cianfrani, M Polverini, in *IEEE INFOCOM*. Keeping the connectivity and saving the energy in the internet (Shanghai, P.R. China, 2011)
46. A Jamakovic, S Uhlig, in *3rd EuroNGI Conference on Next Generation Internet Networks*. On the relationship between the algebraic connectivity and graph's robustness to node and link failures (Trondheim, Norway, 2007). doi:10.1109/NGI.2007.371203
47. JCC Restrepo, CG Gruber, CM Machuca, in *IEEE GreenComm*. Energy Profile Aware Routing (Dresden, Germany, 2009)
48. DW Hong, CS Hong, GH Lee, M-CSPF A scalable CSPF routing scheme with multiple QoS constraints for MPLS traffic engineering. *ETRI J.* **27**(6), 733–746 (2005)
49. N Wang, K Ho, G Pavlou, M Howarth, An overview of routing optimization for internet traffic engineering. *IEEE Commun. Surv. Tutor.* **10**(1), 36–56 (2008). doi:10.1109/COMST.2008.4483669
50. D Adami, DC Callegari, S Giordano, M Pagano, in *IEEE ICC*. A new path computation algorithm and its implementation in NS2 (Glasgow, Scotland, 2007). doi:10.1109/ICC.2007.94
51. Z Wang, J Crowcroft, Quality-of-service routing for supporting multimedia applications. *IEEE JSAC.* **14**(7), 1228–1234 (1996)
52. JC de Oliveira, F Martinelli, C Scoglio, in *IEEE GLOBECOM*. SPeCRA: A stochastic performance comparison routing algorithm for LSP setup in MPLS networks (Taipei, Taiwan, 2002)
53. K Hao, Z Jin, in *IEEE WICoM*. An on-line routing algorithm based on the off-line optimal computing in MPLS (Beijing, China, 2009). doi:10.1109/WICOM.2009.5301929
54. S Nelakuditi, ZL Zhang, *Localized Quality of Service Routing for the Internet*, vol. 739. (Springer, 2003). doi:10.1007/978-1-4615-0383-5
55. A Elwalid, C Jin, S Low, I Widjaja, in *IEEE INFOCOM*. MATE: MPLS adaptive traffic engineering (Anchorage, Alaska, 2001)
56. A Vasilakos, C Ricudis, K Anagnostakis, W Pedrycz, A Pittsillides, in *IEEE International Conference on Fuzzy Systems*. Evolutionary-fuzzy prediction for strategic QoS routing in broadband networks (Anchorage, Alaska, USA, 1998)
57. M Pioro, M Deepankar, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. (Elsevier, 2004). ISBN: 978-0-12-557189-0
58. E Bouillet, G Ellinas, JF Labourdette, R Ramamurthy, *Path Routing in Mesh Optical Networks*. (Wiley, 2007). ISBN: 978-0-470-01565-0
59. AMCA Koster, M Kutschka, in *Proceedings of Photonic Networks*. Network design under demand uncertainties: A case study on the abilene and GEANT network data (VDE Leipzig, Germany, 2011)
60. S Knight, HX Nguyen, N Falkner, R Bowden, M Roughan, The internet topology zoo. *IEEE JSAC.* **29**(9), 1765–1775 (2011). doi:10.1109/JSAC.2011.111002
61. (The MathWorks Inc., Natick, Massachusetts, 2011)
62. IE Kassotakis, ME Markaki, AV Vasilakos, A hybrid genetic approach for channel reuse in multiple access telecommunication networks. *IEEE J. Selected Areas Commun.* **18**(2), 234–244 (2000)
63. Y Yao, Q Cao, AV Vasilakos, EDAL: An energy-efficient, delay-aware and lifetime balancing data collection protocol for heterogeneous wireless sensor networks. *IEEE/ACM Trans. Netw.* **23**(3), 810–823 (2015)
64. P Li, S Guo, S Yu, AV Vasilakos, in *IEEE INFOCOM*. Codepipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding (Orlando, FL, USA, 2012)
65. ZM Fadlulah, T Talen, AV Vasilakos, M Guizani, N Kato, DTRAB: Combating against attacks on encrypted protocols through feature analysis. *IEEE/ACM Trans. Netw.* **18**(4), 1234–1247 (2010)
66. J Esch, A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proc. IEEE.* **100**(12), 3170–3171 (2012)

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)