Toward Energy-Aware Traffic Engineering in Intra-Domain IP Networks Using Heuristic and Meta-Heuristics Approaches

Mojtaba Sabahi Aziz
Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran
s3b3hi@gmail.com
Sepideh Zarei
Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran
sepideh.zarei08@gmail.com
Muharram Mansoorizadeh*
Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran
mansoorm@basu.ac.ir, muharram@gmail.com
Mohammad Nassiri
Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran
m.nassiri@basu.ac.ir

Received: 02/06/2018 Revised: 20/08/2018 Accepted: 21/08/2018

Abstract

Because of various ecological, environmental, and economic issues, energy efficient networking has been a subject of interest in recent years. In a typical backbone network, all the routers and their ports are always active and consume energy. Average link utilization in internet service providers is about 30-40%. Energy-aware traffic engineering aims to change routing algorithms so that low utilized links would be deactivated and their load would be distributed over other routes. As a consequence, by turning off these links and their respective devices and ports, network energy consumption is significantly decreased. In this paper, we propose four algorithms for energy-aware traffic engineering in intra-domain networks. Sequential Link Elimination (SLE) removes links based on their role in maximum network utilization. As a heuristic method, Extended Minimum Spanning Tree (EMST) uses minimum spanning trees to eliminate redundant links and nodes. Energy-aware DAMOTE (EAD) is another heuristic method that turns off links with low utilization. The fourth approach is based on genetic algorithms that randomly search for feasible network architectures in a potentially huge solution space. Evaluation results on Abilene network with real traffic matrix indicate that about 35% saving can be obtained by turning off underutilized links and routers on off-peak hours with respect to QoS. Furthermore, experiments with GA confirm that a subset of links and core nodes with respect to QoS can be switched off when traffic is in its off-peak periods, and hence energy can be saved up to 37%.

Keywords: Energy-aware Traffic Engineering; Green Networking; Greedy Algorithms; Genetic Algorithms.

1. Introduction

The Internet is expanding very fast. Reports generated in 2007 indicate that about 5.5% of whole world energy consumption is related to the Internet, and this number is annually increased by the rate of 20-25% [1]. In 2012 American's home equipment such as modems, routers, and gateways consumed about 803 TWH electricity and produced five million tons of CO_2.

Efforts toward power consumption management in networks cover a wide variety of methods ranging from energy-proportional routing to moving data centers to geographical locations that offer low-cost and/or nature-friendly electricity. Recently green networking solutions [2] were presented with the aim of reducing CO_2 and energy cost by designing energy-aware protocols and planning and manufacturing low power devices.

In traditional networks, routing takes place with static parameters in which packets of data select shortest paths to their destination. Traffic engineering is the task of routing network traffics in an efficient and reliable manner. It uses link load variations as routing parameters in building paths with the aim of optimizing an objective function. Basically, IP routing protocols try to route traffics over shortest paths, considering link weights as a metric. However, focusing on the optimal path may lead to congestion on links constituting the path. Traffic engineering methods try to re-shape and balance the load so that links and paths in other parts of the network are also involved. The main objective here is to keep link utilization in predefined bounds [3]. Energy-aware traffic engineering (EATE) extends the concerns to new dimensions related to power consumption, its costs, and side-effects [4].

As a subfield of green networking, EATE considers energy consumption in routers as the main parameter in routing decision. According to recent statistics, link utilization in networks providing Internet is about 30-40% [5]. Although green networking has attracted lots of attention in recent years, few works have been done on this subject [6]. Low utilized network links, lack of preferable energy management methods for network infrastructures,

* Corresponding Author

increasing cost of energy, increasing number of Internet subscribers, and increasing number of ISPs, are motivations for developing energy-aware traffic engineering approaches.

Turning off network elements to save energy is a key insight in developing energy-aware solutions; however, selection of the elements to turn off is not a trivial task. For a fixed topology and known traffic demand, several subsets of the network elements can be candidates for deactivation. Identifying the best candidate is computationally very expensive, and hence is not feasible for practical applications. Looking for feasible sub-optimal solutions by using approximation algorithms such as greedy approaches, search techniques enriched by heuristics, and random search techniques would be a natural decision for this problem.

Another consideration in the decision for turning off an element is its effect on network stability. For a given traffic demand, an approach can nominate a low utilized link for removal but the link may be required in a slightly modified traffic pattern. These subsequent changes to the network architecture make it unstable and produce management overhead. Hence, any modification to the network architecture must take future traffic variations into account. Estimating future traffic pattern can help in selecting suitable removal candidates. The effectiveness of the approach is directly related to the precision of the estimation process.

In the context of intra-domain traffic engineering, this paper proposes four algorithms for efficient tailoring of the network. The algorithms turn off a subset of network elements and produce a set of paths to transfer the demanded traffic. Selection procedure considers both energy consumption and network stability. Since there are multiple paths from a source to a destination and the paths are usually low-utilized, their traffics can be aggregated and routed over a single path. Elements of the other paths can be deactivated to save energy. It is important to notice that deactivation of links and nodes should not degrade network performance metrics such as maximum link utilization, packet delay, and reliability.

Our main contribution is twofold. First, our approach separates topology management from the routing decision. In this step, we do not alter settings of route setup procedures, link weights, and hyper-parameters. Secondly, we propose a set of new heuristics for sleeping nodes and links. The heuristics are based on the out of the box information from routing infrastructure.

1.1 Preliminaries

In this section, energy usage model of network routers is described. Furthermore, since our approach is based on DAMOTE (Decentralized Agent for MPLS Online Traffic Engineering) algorithm [7], [8] we introduce the algorithm and emphasize its important features.

1.1.1 Energy Usage in Network's Hardware Components

A network router is the main hardware device for traffic engineering. Network routers are composed of a chassis and a number of line cards which have deactivation capability. Line cards are the most important energy users in a router. For instance, 43% of energy usage in a Cisco 12000 router[1] is in its line cards.

We assume that a router has a single line card and each line card has 12 ports. A network link connects two distinct router ports. Line cards consume 40 watts and each port, when it is idle, consumes two watts. Additional 1.73 watts is consumed when a port transmits data with its whole potency. The maximum power consumption of each port will be 3.37 watts. Every port in this paper has 1Gbps bit rate. Comprehensive analysis of network power consumption is discussed in [9]. When all ports of a line card are inactive, the whole line card will be deactivated to save more power. Power usage of a router is calculated by Eq. 1 [9].

Eq. 1 $$P_r = P_{ch} + N_{ln} \times P_{ln} + \sum_{i=0}^{K} (UF \times P_i \times \mathrm{NP}_i) + N_p \times C$$

Where, P_r *is* total power usage in the router and P_{ch} is power consumption in its chassis. N_{ln} and N_p indicate the number of line cards and ports respectively. K is the number of different port configurations. UF is port utilization factor and C is the constant power usage of each port regardless of traffic crossing over it.

1.1.2 Network Energy Proportionality Index

In Figure 1 power consumed by a device is plotted against the load on the device (in Gbps or active number of ports). Ideally, the power consumed should be proportional to the load, with the maximum power consumed, M, being as low as possible. The *ideal* curve represents this desired behavior. In practice, the behaviour of network devices follows the *measured* curve in which the device consumes at least *I* watts. The difference between ideal and measured curves forms the basis of the following energy proportionality index (EPI) for networking components:

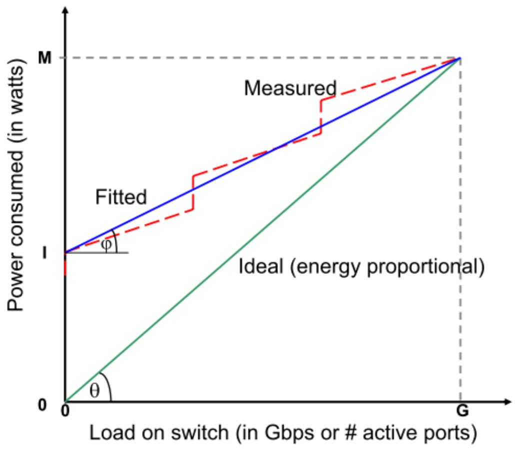

Fig. 1. Ideal and measured power consumption of network devices [9]

Eq. 2
$$EPI = (M - I)/M \times 100$$
$$EPI = \tan\theta / \tan\varphi \times 100$$
$$Normalized\ Power = M/G$$

1. http://www.cisco.com/c/en/us/products/routers/ 12000-series-routers/index.html

If θ and φ are the angles at the origin for the ideal and measured power, EPI is simply $(\tan(\varphi)/\tan(\theta)) * 100$. We express EPI in percentages, with 100 implying that the device has perfect energy proportionality and 0 implying that the energy consumed by the device is always a constant value. Note that EPI is independent of the maximum load that can be carried by the device. Thus, it is most useful in comparison of energy proportionality of devices in the same class. Furthermore, normalized power is the maximum power consumed by the device, *M*, divided by its aggregated bandwidth, *G*. Equation calculates the energy consumption of a router without taking into account the basic power usage, *I*. Actually, energy consumption of transferring one megabit is calculated by equation 3:

Eq. 3
$$Energy_{1-mbps} = (M - I)/G$$

This value is used in estimating power consumption of a network topology serving a traffic demand.

1.1.3 The DAMOTE Algorithm

Basic routing function in this paper uses DAMOTE algorithm. DAMOTE is an MPLS routing function that sets up LSPs online and incrementally in order to optimize an objective function. It uses Bellman-Kalaba shortest path algorithm in building up routing paths.

Different objective functions have been used in DAMOTE of which we focus on the following one that combines load balancing and traffic minimization, as in Eq 4. Here, α allows tuning the trade-off between load balancing and traffic minimization. Lower α will favor longer paths and smooth the load throughout of the network, whereas a greater α will try to minimize the traffic over links. In the case of α = 0, it gives a low blocking probability by avoiding single point of failure. The ordered pair *(i,j)* indicates a link of the set *U*. $X_{(i,j)}^{aux}$ denotes the amount of reserved bandwidth on the link. $W_{(i,j)}^{cap}$ denotes capacity of the link and $\frac{\overline{X^{aux}}}{W^{cap}}$ is the average reserved capacity of all links.

Eq. 4
$$\sum_{(i,j)\in U} \left(\frac{X_{(i,j)}^{aux}}{W_{(i,j)}^{cap}} - \frac{\overline{X^{aux}}}{W^{cap}}\right)^2 + \alpha \sum_{(i,j)\in U} \left(\frac{X_{(i,j)}^{aux}}{W_{(i,j)}^{cap}}\right)^2$$
$$\text{with } \frac{\overline{X^{aux}}}{W^{cap}} = \frac{1}{|U|} \sum_{(i,j)\in U} \frac{X_{(i,j)}^{aux}}{W_{(i,j)}^{cap}}$$

The rest of this paper is organized as follows. Section two gives a brief overview of the related recent works. Section three presents motivation, and proposed algorithms for energy-aware traffic engineering. Analysis of the experimental results is presented in section four. Section five contains some concluding remarks and draws a few future directions.

2. Related Work

Energy-aware traffic engineering has been the subject of many works in recent years. For a thorough and extensive review of the literature see [2], [4], [10], and [11]. Existing methods can be grouped into three main categories, namely *rate adaptation*, *infrastructure sleeping*, and *green networking* [4]. Following the subject of this research, sleep based approaches are reviewed in more detail.

Earlier efforts tried to shed a light on theoretical aspects and proposed several performance measures feasibility analysis. Green-TE is the first energy-aware traffic engineering algorithm [12]. Its basic function is based on a centralized coordinator adopting traffic engineering decisions. In order to reach reliability, this coordinator can be replicated in different locations over the network. Coordinator's responsibility is to gather information from routers, solve the Green-TE problem, and obtain a configuration, and then distribute the decisions to the routers. Based on these decisions a router deactivates some or all of its ports. This method could reduce 27-42% of energy usage. In order to deal with failures, in [13] energy-aware mechanism uses two link-disjoint paths to route traffic demands.

Another approach is proposed in [14] for IP networks. This approach deactivates line cards and chassis of the router to save energy. It consists of three main phases. In the first phase traffic matrices are sorted in decreasing order of the whole traffic demand. The second phase tries to reduce energy consumption by solving a linear optimization problem. The last phase manages congestion while preserving energy consumption within the predetermined bounds.

The approach proposed in [15] is an energy-aware solution for backbone networks. They try to turn off an unused subset of line cards constituting a logical link. The approach could reduce energy consumption by 79% in some cases.

Amaldi et al. [16] modeled and discussed the energy-aware routing problem as a *Mixed Integer Linear Programming (MILP)* optimization. A deterministic solution for this problem can be found by search techniques such as backtracking and branch-and-bound. However, these techniques may require an exponential amount of time and memory in worst case scenarios. Since the problem is NP-hard, they proposed heuristic approaches based on *Interior Gateway Protocol Weight Optimization (IG-WO)* to find feasible sub-optimal solutions. Their approach is composed of greedy search steps that exploit properties of IG-WO.

Ruiz-Rivera et al. [17] Studied the problem of reducing energy consumption in MPLS networks. They compared online and offline heuristics for LSP setup; concluding that for well-known topologies such as Abilene and AT&T it is desirable to achieve LSP acceptance rates above 90% with up to 20% of links shut down.

Fortz reviews meta-heuristic approaches for traffic engineering in IP networks [18]. The studied works mostly try to set link weights in OSPF based routing. The goals of these works are controlling congestion and finding low-cost routes. Few of the studied works also consider node failures and link breakdowns. Compared to the context of energy-aware routing, these goals are short-term and highly dynamic. While the mentioned works proved to be efficient in terms of finding proper routes, their real-time application remains a challenge.

An approach based on genetic algorithms (GAs) is proposed in [19] to reduce network energy consumption besides reducing network reconfiguration rate. GA is a meta-heuristic algorithm in which an initial generation evolves by passing through some generations. Every member in each generation keeps topology information. A member is feasible when it can transfer all the traffic demands and at the same time satisfy maximum power consumption constraint. A member's fitness value is the weighted sum of normalized power consumption and reconfiguration costs. Samadi et al. [20] adopted *Non-dominated Sorting Genetic Algorithm II (NSGA-II)* for load balancing and energy consumption management. Their multi-objective approach induces sub-optimal Pareto-fronts as solutions.

In summary, existing works model the energy-aware traffic engineering as NP-hard mixed integer programming problem, and hence try to induce near-optimal solutions by means of various heuristics and metaheuristics. The approaches vary from self-sleeping routers to inter-layer energy-aware protocols. However, there is a strong agreement that to be applicable in practical scenarios, a solution must be used on top of existing OSPF and MPLS based networks. As an important challenge, solutions that exploit protocol-specific features (e.g. link weights in route setup procedure) are tightly coupled to the special protocol that limits their application and flexibility. For example, a solution that is developed on IGP-WO link weights needs to be redesigned for MPLS networks. Furthermore, its modifications to link weights may interfere with other network tasks such as QoS enforcement operations.

3. Proposed Approach

Theoretically, a network with *M* nodes and *N* links may have 2^{M+N} different topologies. As discussed in related works, (e.g. [2] and [16]), Finding the best topology in this huge space is an NP-hard problem. Hence, instead of trying to find the optimal solution, one can try to find feasible sub-optimal solutions in a reasonable amount of time and computation resources. Various techniques have been developed for this sub-optimal searching problem. These techniques are broadly classified into three main categories:

- **Graph Traversal:** graph traversal algorithms model the solution space as a graph that its nodes are candidate solutions. There is a link between nodes A and B if solution A can be transformed to B by a set of pre-defined operations (e.g. by flipping order of nodes). Backtracking and Branch-and-Bound are well-known search algorithms in this category. The former traverses the graph by depth-first order and the later follows breadth-first order. In a worst-case, backtracking may fail after an exponential amount of time without reporting a feasible solution. Similarly, branch-and-bound search may require an exponential amount of memory for queuing intermediate solutions. Due to these limitations, graph traversal algorithms are not suitable for practical scenarios.
- **Greedy Search:** greedy algorithms build solutions gradually by taking locally best steps. Naturally, these algorithms are deterministic and require a polynomial amount of time and memory. The drawback is that greedy algorithms usually converge to local minima that can be far away from optimal solutions.
- **Random search:** Random search techniques try to inspect diverse parts of the solution space. A random search procedure starts with an initial point in the solution space and then iteratively generates new solutions in the neighborhood of the current solution. Genetic algorithms (GA's) are a type of random search technique that combines random selection with biological evolution concepts. For discrete selection problems (as in our case of topology selection) it is known that GA's are better suited than their counterparts such as Particle Swarm Optimization (PSO) based techniques [21].

We assume that traffic demands are stable and smooth at least for a few hours. That is, we do not consider large traffic fluctuations in short time intervals. Because frequent modifications of network structure by turning elements on and off will make it very unstable that is an important challenge by itself and must be addressed separately [22], [23]. However, we try to compensate for small variations by limiting maximum link utilization.

Our main approach is to generate candidate topologies out of available network and then select the best one for traffic delivery. We treat the underlying routing procedure as a black-box element that accepts a network structure and a traffic demand matrix, then tries to route the demand over the network.

As depicted in Figure 2, our approach has two distinct modules. The first module generates topologies by means of heuristic and meta-heuristic algorithms. The evaluation module feeds the generated topology along with the demanded traffic to the routing procedure and grabs its output. The routing procedure outputs routing information such as built paths, employed links, and their occupied capacities. The module then analysis these information and estimates feasibility and usefulness of the topology.

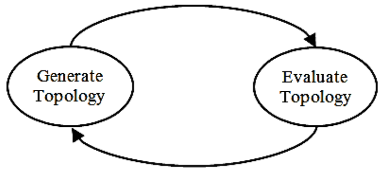

Fig. 2. Main modules of the proposed approach

Based on greedy and random search techniques, we propose four algorithms to employ energy-awareness in routing. The rationale behind these approaches is that simpler networks, e.g. networks with small a number of links and nodes, will consume less energy. Hence, we try to turn off ports, links, and nodes as much as possible. It is quite obvious that the simplified network must serve the demanded traffic as the original network.

3.1 Sequential Link Elimination

Sequential Link Elimination (SLE) is a greedy backward selection procedure (Algorithm 1) that switches off links one by one and evaluates the resulting network. If the network can transfer the demanded traffic and over-all utilization is lowered, the link will be marked as a candidate for inactivation. After processing all of the links, the candidate with minimum utilization will be switched off. This process is repeated until no more links can be deactivated. Finally, nodes with no active links will be turned off to save more energy.

Algorithm 1. Sequential Link Elimination(SLE)

```
Input: Original Network(V,E),
Traffic Demand (D)
Output: Prunned network
finished = false
repeat
  for e ∈ E
    util[e] = max_util(<V,E- e>)
  e = argmin(util)
  if util[e] < threshold
    E = E - e
  else
    finished = true
until finished = true
for v ∈ V
  mark[v] = false
for <u, v> ∈ E
  mark[u] = true
  mark[v] = true
for v ∈ V
  V = V-v iff mark[v] = false
return (<V, E>) // Reduced network
```

3.1.1 Time Complexity

The main operation of this algorithm is calling DAMOTE. Suppose that the network under consideration has *N* active links and *M* nodes. In each iteration, the algorithm will deactivate a link or terminate. In worst case, DAMOTE is called $N + (N-1) + \ldots + 1$ times. which belongs to $O(N^2)$. Combined with DAMOTE's running time, O(NM), total complexity of the algorithm is $O(N^3M)$.

A simple improvement to the SLE would be applying *maximum utilization constraint* that does not allow links to use more than 80 percent of their capacity. This constraint prevents turning off some links and leads to less energy saving but more robust topology against demand variations.

3.2 Extended Minimum Spanning Tree

For an undirected graph, a spanning tree is a tree composed of the graph's nodes and a subset of its links that preserves connectivity of the original graph. Assuming that each link of the graph has a non-negative real-valued weight, cost of a tree is the sum of weights of its links. A minimum spanning tree (MST) is a spanning tree with the minimum cost. Algorithms such as Prim and Kruskal efficiently build MSTs [24].

An immediate idea for pruning network would be replacing the original network with its MST. The weight of a link connecting two nodes is the minimum number of active ports of these nodes.

A problem of this algorithm is its inability to turn off nodes. Extended minimum spanning tree (EMST) (Algorithm 2) resolves this issue by detecting removable nodes. Suppose that a path is established for a given traffic demand. Source and destination of the path are called *edge* nodes. Other nodes in the path are *core* nodes. In this algorithm, shortest paths between all pairs of edge nodes are calculated. Nodes which are not in any shortest path will be switched off. Core nodes are then sorted according to their frequency of presence in the shortest paths. The node with the lowest presence will be turned off and the shortest paths between every pair of edge nodes will be re-established. If there is a path for each pair of the edge nodes, the algorithm goes on with the next core node in the list. But, if there is a pair with no path, the core node is turned back on and the algorithm terminates. The final topology excluding inactive nodes and links is fed to the minimum spanning tree algorithm.

The topology must be able to transfer traffic demand. If it fails, some nodes and links should be added to it as the post-processing step. This step sequentially adds links to the topology and terminates when the topology can transfer the demand.

Algorithm 2. Extended Minimum Spanning Tree (EMST)

```
Input: Original Network(V,E),
Traffic Demand (D)
Output: Pruned network
P = build_path(<V , E>, D )
for v ∈ V
  f[v] = #{p|p∈P ∧ v∈p}
for v ∈ V
  V = V – v if f[v]= 0
descend_sort(f)
for each v in f
  disable (v)
  if isvalid (<V-v , E>, D)
    T = MST( <V-v , E> )
    if isvalid(T, D)
      V = V - v
    else
      exit for
return <V,E> //pruned network
```

3.2.1 Time Complexity

Suppose that M and N are the number of edge and core nodes, respectively. Using Floyd's all-pairs-shortest-path algorithm, the time complexity of finding shortest paths is $O(M+N)^3$. M^2 is the number of all possible paths. Since each core node is searched in all of the M^2 paths and maximum path length is N, searching is of order $O(M^2N)$. Sorting the list of core nodes takes O(Nlog(N)) time and another $O(M^2)$ time is required for minimum spanning tree construction. Summing up the terms, the algorithm requires $O(M+N)^3$ running time.

3.3 Energy-aware DAMOTE

Energy-aware DAMOTE (EAD) switches off elements of network infrastructure to decrease topology size and save energy and turns them back on when they are required.

The objective of EAD is minimizing maximum link utilization and minimizing energy consumption of network.

At first, DAMOTE algorithm is run on the topology and link utilizations are obtained. Links with zero utilization are then turned off. These links are redundant and can be used in the case of failures. Other links are sorted according to their utilization and energy priority and then sequentially processed for removal (Algorithm 3).

Links are first sorted according to their utilization in a list, L_1, and also according to their energy priority in another list, L_2, in ascending order. The relative order of a link in the combined list, L, is determined by sum of its positions in L_1 and L_2. For instance suppose that for a network with four links, namely {e1, e2, e3, e4}, relative order of utilization is {e1, e2, e4, e3} and relative order of energy is {e4, e3, e2, e1}. Taking into account both factors, the combined order would be {e3, e1, e4, e2}.

Algorithm 3 Energy-aware DAMOTE (EAD)

```
Input: Original Network(V,E),
Traffic Demand (D),
MLU threshold (a)
Output: Pruned network
A = evaluate (<V,E> , D) ;
L1 = sort_by_energy(A, E)
L2 = sort_by_util (A, E)
L = combine(L1,L2)
for e ∈ L
  A = evaluate (<V,E -e> , D) ;
  if isvalid(A)
    if max_util(A) < a
      E = E -e //permanently
    else
      enable(e)
  else
    exit for
```

Each link is marked for removal and the topology excluding it is tested against the traffic matrix. If it can handle the traffic, the link is turned off. Furthermore, nodes with no active links are turned off to save more energy.

A similar argument as EMST applies for EAD, concluding that time complexity of this algorithm is also at most $O(|V|+|E|)^3$

3.4 Genetic Algorithms

Genetic algorithms (GAs) are tools for random search in potentially diverse and huge solution spaces [25]. In GA terminology, a solution is called *an individual*. Basically, a GA procedure starts with an initial set of candidate solutions that is called the *first generation*. Each new generation is induced from earlier generations with the aid of genetic operations. The operations combine multiple individuals or modify a single individual to get better individuals. While GA does not guarantee to find an optimal solution, it usually finds a feasible and good sub-optimal solution.

Formulation of an optimization problem to be solved by GA is composed of several steps. At first, genes and chromosome structures must be built up. Then, procedures for initiating the first generation, evaluation of solutions and induction of subsequent generations must be defined. In the literature, usually, classic GA procedures are adopted with a few extensions for GA operations. In the following subsections, details of using GA for topology pruning is presented.

3.4.1 Genes and Chromosomes

As depicted in Figure 3, a network with M nodes and N links is represented as a chromosome with M+N genes. In this formulation, a gene is a three-state variable denoting presence or absence of the respective node or link. A zero-valued gene denotes that the respective node or link is inactive (disabled or turned off) so that it does not consume energy. A gene with a value of one denotes that the respective node or links is active and consumes energy. A value of two for a gene denotes that corresponding node or link must be permanently active. Permanent nodes and links are identified automatically based on the demanded traffic. That is, source and destination nodes of all the flows are marked as permanent nodes. If a permanent node is connected to the network with just a single link, the link is also marked as permanent to enforce network connectivity.

3.4.2 Initial Population

After marking permanent nodes and links, the initial population is generated by random setting of non-permanent genes. This stage requires population size which is usually set by the user.

3.4.3 Evaluation

Usefulness or *fitness* of a solution is proportional to its energy saving. Eq. 5 calculates fitness value; *f;* for a given solution; x. E_{old} is the energy that is consumed by the basic topology. E_{new} is the energy consumption of the modified topology. Solutions that fail to service all the demanded traffic or violate maximum utilization constraint are ignored.

Eq 1.

$$f(x) = \begin{cases} 1 - \left(\frac{E_{new}}{E_{old}}\right), & if \max_{util} < 0.8 \\ 0, & otherwise \end{cases}$$

3.4.4 Selection and Reproduction

The concept of evolution in GA starts with the *selection* procedure in which, two members of the current generation are selected for reproduction. These members are called parents and the newly generated individuals are called *off-springs* or *children*; analogous to reproduction procedure in most biological systems. We use well-known roulette-wheel selection method that assigns selection probabilities proportional to its fitness values.

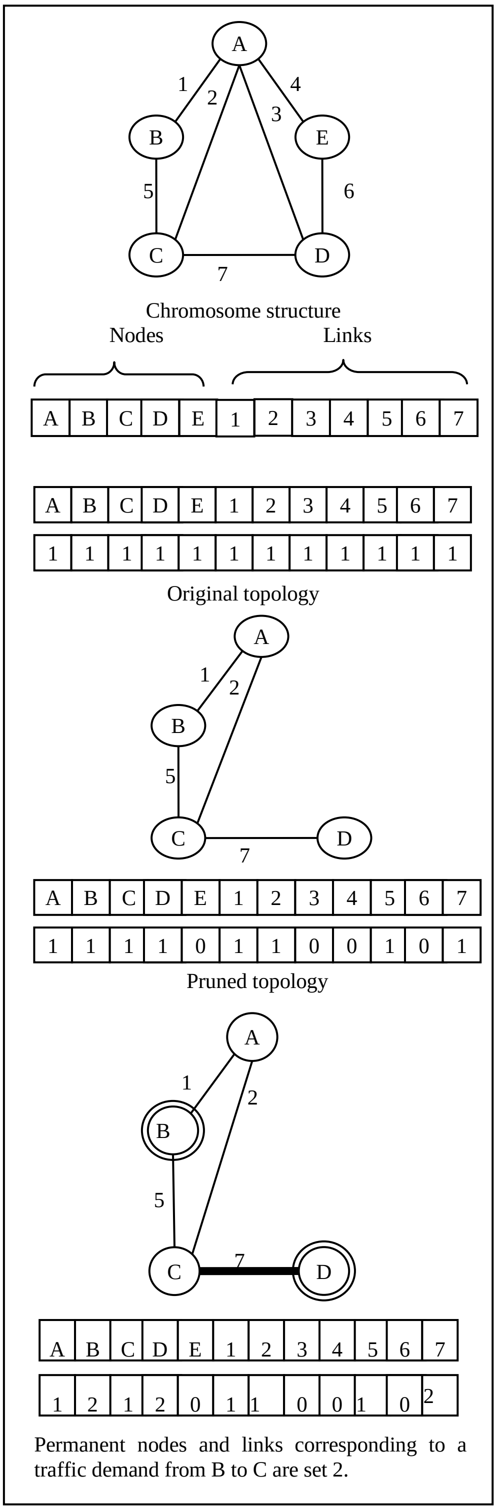

Fig. 3. GA representation of a network

A reproduction operation is a function that gets an individual or two individuals as input and outputs a set of new individuals. Usually, two types of reproduction operations are used. A *mutation* function alters a single gene of an individual. Mutation introduces more diversity into the population. *Crossover* function exchanges portions of two individuals and generates their children in the hope the children are more feasible than their parents. We use two-point cross-over and single-point mutation, as depicted in Figure 4.

3.4.5 Stopping Criteria

Selection and reproduction operations are repeated several times over the current generation to induce a new generation. Each subsequent generation is built with the hope that its members would be better than the previous generation.

The process of evolution through generations stops when some set of sufficiently good individuals are found or there is a little or no hope to find suitable individuals in the future.

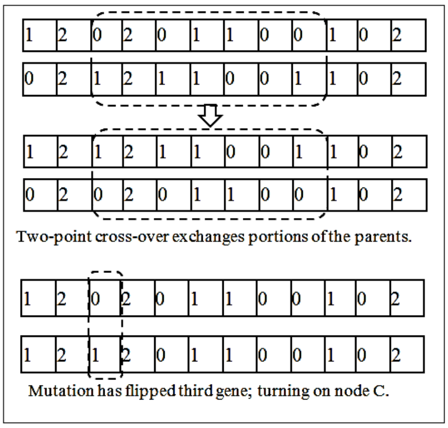

Fig. 4. GA evolutionary operations

4. Simulation and Performance Evaluation

Proposed algorithms are implemented in C-language, encapsulating TOTEM toolbox's DAMOTE algorithm[1] [7]. TOTEM is an open source toolbox for modeling, simulation, and evaluation of protocols and algorithms for network traffic engineering. DAMOTE is a traffic engineering package featuring a configurable score function and rich report generation. Among various quantitative and qualitative attributes of topologies, statistics such as *maximum links utilization*, *total sum of network energy consumption,* and number *of inactive links and nodes* are of particular interest in this research.

For genetic algorithms, we used populations with 1000 members evolving through 20 generations. Cross-over rate was 1, since we were to generate a population of predefined size, and mutation rate was 0.01.

4.1 Evaluation Information

Abilene topology and its traffic matrices were used in experiments [26]. Since all nodes in this topology are edge nodes (i.e. flow sources or destinations), to test deactivation feature of the proposed algorithms an augmented topology is generated by adding virtual core nodes to the original Abilene topology. The new topology is called extended Abilene. Table 1 summarizes specifications of Abilene and Extended Abilene networks.

1. http://totem.run.montefiore.ulg.ac.be/algos/damote.html

Table 1. Specifications of Abilene and Extended Abilene Networks

Name	#Nodes	#Links	#Edge Nodes	#Core Nodes	%Core Nodes
Abilene	12	30	12	0	0
Extended Abilene	21	52	12	9	43

Following the idea of [16], traffic matrices of these networks are scaled by 10 so that maximum link utilization reaches up to 50%. Scaling traffic matrix makes the proposed algorithms more robust to variations in traffic patterns. Figure 5 shows the utilization of links in extended Abilene network at 22:00 on December 15, 2007. It confirms that most link utilizations are below 0.5. For the same network with 10-times scaled traffic matrix, maximum network utilization in different hours of a day is displayed in Figure 6. The figure confirms that maximum network utilization is below 0.6 most of the time. Over-plotted dashed line is the trend line of the maximum network utilization. Its slightly increasing trend confirms that network traffic is not stationary; hence there is no single best network topology for all hours of a day.

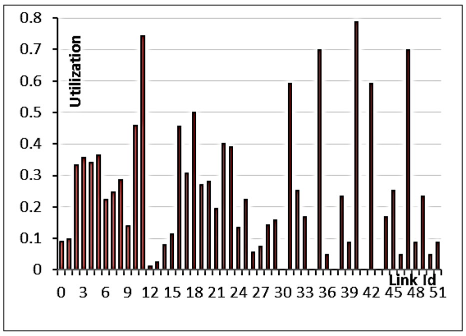

Fig. 5. Link utilization in extended Abilene

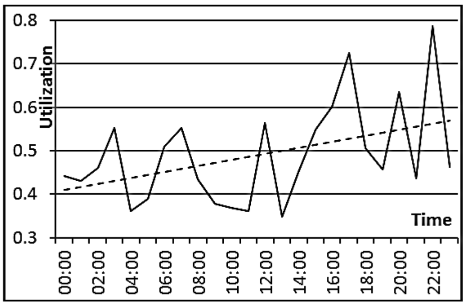

Fig. 6. Actual hourly network utilization for extended Abilene

4.2 Network Links Utilization

The four proposed algorithms are applied to the extended Abilene network and results are reported hereafter. Maximum network utilization during a day is displayed in Figure 7. For clarity of the demonstration, their respective trend lines are plotted in Figure 8. EMST has higher utilization than other algorithms. After that, SLE has a similar trend. EAD has utilization around 0.68 with a constant trend that makes it a stable algorithm. Among all, EAGA has the steepest trend denoting its adaptive behavior against traffic variations.

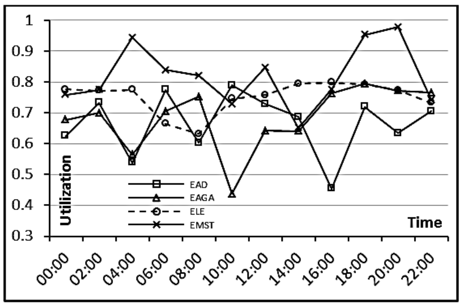

Fig. 7. Hourly network utilization

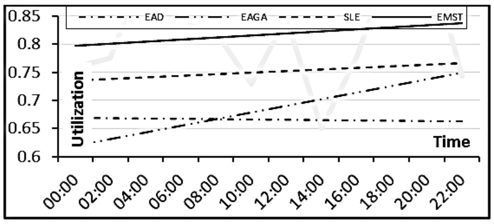

Fig. 8. Trends of network utilization (see fig. 7)

4.3 Power Saving

The amount of power saving of the four algorithms is presented in Figure 9 and Table 2. Figure 10 and Figure 11 report percentage of deactivated links and nodes for various algorithms, respectively.

As before, ELE, EAGA, and EMST have similar patterns in power saving. They achieve almost 38% power saving in most of the hours in a day. EAD has lower saving rate as compared to others. This may be due to the fact that it tries to keep maximum link utilization as low as possible. This requires incorporating more links and nodes, hence using more energy.

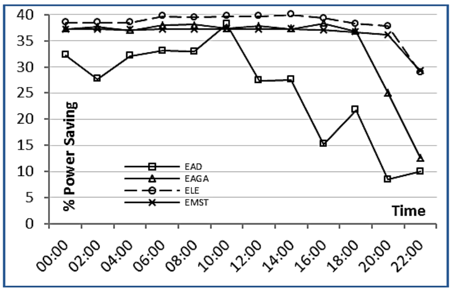

Fig. 9. Power saving of the algorithms

Table 2. Number of deactivated links and nodes

	EAGA		EMST		ELE		EAD	
	N_off	L_off	N_off	L_off	N_off	L_off	N_off	L_off
00:00	6	28	6	28	6	30	5	26
02:00	6	29	6	28	6	30	4	26
04:00	6	27	6	28	6	30	5	26
06:00	6	30	6	28	6	33	5	29
08:00	6	30	6	28	6	33	5	28
10:00	6	28	6	28	6	33	6	31
12:00	6	29	6	28	6	33	4	25
14:00	6	28	6	28	6	34	4	25
16:00	6	31	6	28	4	30	2	16
18:00	6	27	6	27	6	30	3	22
20:00	4	20	6	27	2	21	1	10
22:00	2	10	5	18	3	22	1	14

N_off: #deactivated nodes, L_off: #deactivated links

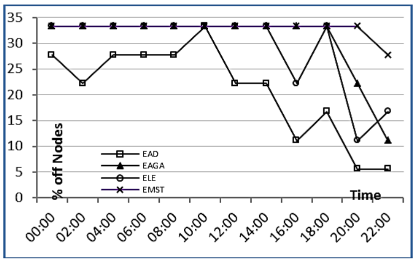

Fig. 10. Percentage of inactive nodes

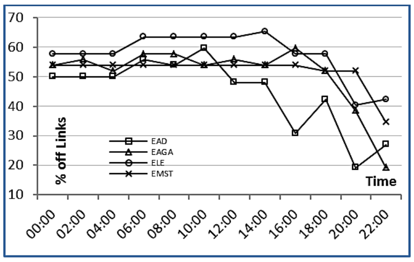

Fig. 11. Percentage of inactive links in extended Abilene

4.4 Comparison with Related Works

We compared the performance of our proposed algorithms against that of two notable recent works, one based on OSPF (Amaldi et al. [16]) and the other one based on MPLS (Ruiz-Rivera et al. [17]).

As mentioned earlier, Amaldi et al. alter link weights in order to direct the routing procedure to find more energy efficient paths. They report their experiments with two traffic matrices over Abilene and extended Abilene networks. One for 6:00 that denotes a low traffic load and the other for 12:00 that represents the highest load.

Table 3. Comparison of our algorithms against Amaldi et al. [16]

	06:00 AM (low traffic)				12:00 AM (high traffic)			
	L-off	N-off	%Saving	%MLU	L-off	N-off	%Saving	%MLU
ELE	33	6	37	66	30	6	37	78
EMST	28	6	37	84	28	6	37	76
EAGA	28	6	38	70	30	6	37	68
EADA	29	5	33	77	26	5	32	62
Amaldi	54	7	54	42	40	3	21	78

Table 3 summarizes results for the compared algorithms. Amaldi et al. works well for low traffic scenarios and turns off much more links as compared to ours. However number of turned off nodes are similar. For heavily loaded scenarios, our algorithms work much better than Amaldi's.

Ruiz-Rivera et al. enhance energy consumption in MPLS networks by favoring LSPs that share links with previous or future requests. They report link utilization for various levels of bound width requests. As the required bandwidth increases, maximum link utilization increases as well. Since their experimental setup and results format is significantly different from ours (and also from Amaldi et al.) a comprehensive comparison and discussion of the results is not applicable. However, we compare the rate at which link utilization increases in response to the traffic increments. Table 4 reports mean and standard deviation of maximum link utilization for studied methods. For lower traffics, link utilization is low. Smaller mean values denote that the method keeps MLU at lower levels by incorporating more links. This results in more energy consumption in the network. Hence, lower MLU roughly translates to lower energy saving. With this observation, ELE and Ruiz-Rivera seem to be more energy efficient than others.

Focusing on MLU deviations, lower deviation means that the algorithm uses links in accordance with requested traffic load. That is, if the traffic increases, more links are employed to keep MLU within bounds. However, higher deviation means that the set of active links are decided in advance and their utilization varies in response to request variations. With this observation, it can be concluded that Ruiz-Rivera uses many lightly loaded links for low-traffic scenarios. That is, the approach saves more energy for heavy traffic scenarios but is not energy efficient for lightly loaded cases.

Table 4. Maximum link utilization for compared algorithms

	ELE	EMST	EAGA	EADA	Amaldi	Ruiz
Mean	78	82	68	67	65	79
Std.	5	10	10	10	15	24

4.5 Adaptive Topology Planning

The ultimate goal of developing these algorithms is planning and proposing a topology for a network that is more energy-efficient than the original network. The proposed topology must be robust to traffic fluctuations and save energy as much as possible. The central idea in planning is to define a profile as a set of conditions for the network. Then, a topology is constructed and archived for each network profile. This set is built offline using the proposed algorithms. Being offline allows us to employ sophisticated simulation and estimation techniques without much worrying about the time complexities.

As a simple implementation of this approach, we consider temporal profiles. As mentioned earlier, traffic matrices used in these experiments are for 12 two-hour intervals, namely 00 to 02, 02 to 04, ..., 22 to 00. Using any of the proposed algorithms the best topology is constructed for each interval. This topology has a set of links and nodes that are subject to deactivation. To make smooth changes to the network, and hence make it more stable, a day is partitioned into three equal intervals, namely 0 to 8, 8 to 16, and 16 to 0. Each of these eight-hour intervals spans four two-hour intervals for which there exists a special topology. Then, a topology is proposed for eight-hour intervals that excludes inactive nodes and links of all the corresponding four topologies.

Profile-based planning enables dynamic selection of proper topology for current network status. If a traffic demand cannot be handled by the current topology, another topology with more bandwidth resources will be enabled in which more links and/or nodes are active. The new profile can be switched proactively by estimation of network status in the near future. For example, if current maximum link utilization goes beyond 90% a profile with more resources can be used to prevent possible future failures.

5. Conclusion and Future Directions

In the scope of green networking, this paper proposed four algorithms for adaptive and dynamic selection of topologies for backbone data networks. These algorithms try to tailor down the network so that besides serving demanded traffic, a set of its nodes and links are deactivated to save energy.

The proposed algorithms demonstrated feasibility and usefulness of automatic topology adaptation. Sequential Link Elimination (SLE), Extended Minimum Spanning Tree (EMST), and Energy-Aware DAMOTE (EAD) are deterministic algorithms that serially process nodes and links for removal. SLE works in backward manner in which at first all the links and nodes are engaged. In subsequent steps, it tries to turn off links while preserving network status in serving all the demanded traffic. EMST takes a reverse approach by constructing minimum spanning tree of the network. Initially, all the links out of this tree are inactive. The algorithm adds links to the tree until it can handle the demanded traffic. EAD is similar to SLE in the sense that it sequentially processes the links. The difference is that in EAD, relative order of links is determined by both their energy saving and maximum utilization features.

SLE, EMST, and EAD are greedy algorithms. Greedy algorithms usually find a suboptimal solution that can be very different from optimal solutions. On the other hand, Genetic Algorithms (GAs) randomly search solution space for proper suboptimal solutions. Usually, a GA with a sufficiently large population size and generations finds near-optimal solutions. However, the computational cost of running GAs is more than the deterministic algorithms.

Experiments on extended Abilene network confirmed that the proposed algorithms are capable of saving a considerable amount of energy consumption in the network. Analysis revealed that EAD is best in controlling congestion and EAGA is better in saving more energy.

In the future, current research can be extended in several dimensions. At first, a distributed and online solution in which each core node reactively decides by itself to enable or disable its outgoing links is desirable. The node inspects the current network status and performs an action to make it better. Reinforcement learning based approaches can model this behavior very efficiently. However, it would require plenty of training network history to make good decisions.

This paper has focused on the net amount of energy that is used by the network. Consider a network that some of its elements use solar energy and some others use fossil-fuel electricity. An extension of current research would be taking to account natural preferences of these sources (e.g. solar to fossils).

Genetic algorithms are computation intensive by nature. Recently, parallel implementation of GAs on multiprocessor systems or special purpose boards such ad graphical processing units (GPUs) has caught a considerable interest. Online implementation of GAs on newer technologies and modeling complex and feature-rich networks by GAs is another direction for future research.

References

[1] J. Baliga, K. Hinton and R. S. Tucker, Energy consumption of the Internet, University of Melbourne, Department of Electrical and Electronic Engineering, 2011.

[2] C. Fang, F. R. Yu, T. Huang, J. Liu and Y. Liu, "A Survey of Green Information-Centric Networking: Research Issues and Challenges.," IEEE Communications Surveys and Tutorials, vol. 17, pp. 1455-1472, 2015.

[3] I. F. Akyildiz, A. Lee, P. Wang, M. Luo and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," Computer Networks, vol. 71, pp. 1-30, 2014.

[4] A. G. Cervero, M. Chincoli, L. Dittmann, A. Fischer, A. E. Garcia, J. Gal{\'a}n-Jim{\'e}nez, L. Lefevre, H. d. Meer, T. Monteil, P. Monti and others, "Green wired networks," Large-Scale Distributed Systems and Energy Efficiency: A holistic view, pp. 41-80, 2015.

[5] M. Baldi and Y. Ofek, "Time for a" greener" internet," in Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on, 2009.

[6] R. Bolla, R. Bruschi, F. Davoli and F. Cucchietti, "Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures," Communications Surveys \& Tutorials, IEEE, vol. 13, pp. 223-244, 2011.

[7] G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D'Arienzo, O. Delcourt, J. Domingo-Pascual, S. Cerav-Erbas, I. Gojmerac, X. Masip and others, "An open source traffic engineering toolbox," Computer Communications, vol. 29, pp. 593-610, 2006.

[8] F. Blanchy, L. M{\'e}lon and G. Leduc, "A Preemption-aware on-line routing algorithm for MPLS networks," Telecommunication Systems, vol. 24, pp. 187-206, 2003.

[9] P. Mahadevan, P. Sharma, S. Banerjee and P. Ranganathan, "A power benchmarking framework for network devices," in NETWORKING 2009, Springer, 2009, pp. 795-808.

[10] K. Cengiz and T. Dag, "A review on the recent energy-efficient approaches for the Internet protocol stack," EURASIP Journal on Wireless Communications and Networking, vol. 2015, pp. 1-22, 2015.

[11] A. P. Bianzino, C. Chaudet, D. Rossi and J. L. Rougier, "A Survey of Green Networking Research," IEEE Communications Surveys Tutorials, vol. 14, pp. 3-20, 2012.

[12] M. Xia, M. Tornatore, Y. Zhang, P. Chowdhury, C. U. Martel and B. Mukherjee, "Greening the Optical Backbone Network: A Traffic Engineering Approach.," in ICC, 2010.

[13] G. Lin, S. Soh, M. Lazarescu and K.-W. Chin, "Energy-aware two link-disjoint paths routing," in High Performance Switching and Routing (HPSR), 2013 IEEE 14th International Conference on, 2013.

[14] B. Addis, A. Capone, G. Carello, L. G. Gianoli and B. Sanso, "Energy-aware multiperiod traffic engineering with flow-based routing," in Communications (ICC), 2012 IEEE International Conference on, 2012.

[15] W. Fisher, M. Suchara and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off

cables in bundled links," in Proceedings of the first ACM SIGCOMM workshop on Green networking, 2010.
[16] E. Amaldi, A. Capone and L. G. Gianoli, "Energy-aware IP traffic engineering with shortest path routing," Computer Networks, vol. 57, pp. 1503-1517, 2013.
[17] A. Ruiz-Rivera, K.-W. Chin, S. Soh and R. Raad, "On the performance of online and offline green path establishment techniques," EURASIP Journal on Wireless Communications and Networking, vol. 2015, pp. 1-17, 2015.
[18] B. Fortz, "Applications of meta-heuristics to traffic engineering in IP networks," International transactions in operational research, vol. 18, pp. 131-147, 2011.
[19] E. Bonetto, L. Chiaraviglio, F. Idzikowski and E. Le Rouzic, "Algorithms for the multi-period power-aware logical topology design with reconfiguration costs," Journal of Optical Communications and Networking, vol. 5, pp. 394-410, 2013.
[20] R. Samadi, M. Nassiri and M. Mansoorizadeh, "Energy-aware traffic engineering in IP networks using non-dominated sorting genetic II algorithm," International Journal of Advanced Intelligence Paradigms, vol. 1, 2019.
[21] R. Hassan, B. Cohanim, O. De Weck and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in 46th AIAA/ ASME/ ASCE/ AHS/ ASC structures, structural dynamics and materials conference, 2005.
[22] O. Okonor, N. Wang, S. Georgoulas and Z. Sun, "Green Link Weights for Disruption-Free Energy-Aware Traffic Engineering," IEEE Systems Journal, vol. 11, pp. 661-672, 2017.
[23] R. Wang, S. Gao, W. Yang and Z. Jiang, "Energy aware routing with link disjoint backup paths," Computer Networks, vol. 115, pp. 42-53, 2017.
[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to algorithms third edition, MIT Press, 2009.
[25] A. P. Engelbrecht, Computational intelligence: an introduction, John Wiley & Sons, 2007.
[26] S. Orlowski, M. Pi{\'o}ro, A. Tomaszewski and R. Wess{\"a}ly, "SNDlib 1.0--Survivable Network Design Library," in Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, 2007.

Mojtaba Sabahi-Aziz Got his BSc from Shamsipour faculty of engineering in software engineering in 2013. He earned his MSc in computer networks from Bu-Ali Sina University in 2016. His research interests include monitoring and evaluation of computer networks.

Sepideh Zarei Got her BSc in software engineering in 2013 and MSc in computer networks in 2016 both from Bu-Ali Sina University. Her research interests include monitoring and evaluation of computer networks.

Muharram Mansoorizadeh is an assistant professor at the Computer Department of Bu-Ali Sina University. He received his BSc degree in software engineering from the University of Isfahan, Isfahan, Iran, in 2001, and his MSc degree in software engineering and the PhD in computer engineering from Tarbiat Modares University, Tehran, Iran, in 2004 and 2010, respectively. His current research interests include machine learning, affective computing and information retrieval.

Mohammad Nassiri is an assistant professor at the Computer Department of Bu-Ali Sina University. He received his BSc and MSc degrees from Iran University of Science and Technology (IUST) and Sharif University of Technology, respectively, in 2000 and 2002. He also received his Ph.D. in Computer Engineering from Grenoble INP, France, in 2008. His current research interests include Performance Evaluation of wireless networks and the Internet of things, Internet Traffic Classification and Software networking paradigm.