Embedding Multiple-Step-Ahead Traffic Prediction in Network Energy Efficiency Problem

Abdolkhalegh Bayati, Kim Khoa Nguyen, Mohamed Cheriet
École de Technologie Supérieure, University of Quebec, Canada
Email: {Abdolkhalegh.Bayati, knguyen, Mohamed.Cheriet}@synchromedia.ca

***Abstract*—Adaptive Link Rate (ALR) is widely used to save energy consumption of network by adjusting the link rate according to the carried traffic through a network-level optimization of the flow allocation process. Existing ALR solution is mainly reactive, in which link speed is changed only when new traffic demand is requested. Also, they focus on energy consumption, and do not consider the cost of changes in the network (e.g., change in traffic routes, and link rates). Once bandwidth has been allocated for a demand, the link rate remains constant during the entire session. Therefore, this solution may result in sub-optimal schemes and requires multiple re-optimizations as traffic flows are fluctuating during the session, hence reducing the overall network performance. In this paper, we improve the ALR with a multiple-step-ahead method to optimize link rates based on forecasting traffic demand predictively. We formulate the proposed Predictive ALR (PALR) as an Integer Linear Programming (ILP) model and then design a heuristic simulated annealing (SA) -based algorithm to solve it. Our experimental results show our approach provides energy saving while it decreases on average 18% of link state transition and 11% of the flow reroutings compared to the original ALR.**

***Index Terms*—Adaptive Link Rate (ALR), Multiple-Step-Ahead Traffic Prediction, Energy-Aware Routing (EAR)**

I. Introduction

Energy consumption of the communication systems contributes a considerable fraction of networking costs and environmental-related concerns [1]. The network power consumption is independent of the offered traffic load and remains high even during underutilized periods [2]. Power proportionality in network devices is a technique to address this problem [3]. A network element is said to be energy-proportional if its power consumption is proportional to its carried traffic. There are two main approaches to providing energy proportionality [3]: (i) deactivating (or powering off) the device during idle periods, and (ii) using adaptive link rate (ALR). In the first approach, data is sent faster in the active interval to have a longer sleep interval. In the second approach, link speed is decreased when the link is underutilized. It is based on the fact that the interface cards with lower transmission rates consume smaller amounts of power [4].

ALR has been employed in energy-aware routing (EAR) methods to optimize the network energy consumption [2]. In ALR-based EAR, the network resources and link capacities are adapted to the traffic load to ensure energy conservation. It forms a mixed integer programming (MIP) optimization problem with energy consumption as its objective function [4]. The control variables are the paths of the flows and states of the links. It examines the possible flow allocations to minimize power consumption while guarantees the network performance. This centralized approach uses the global network and traffic information available through the network controller as input to the optimization process.

In networks with static nonvarying traffic, the instance of ALR-based EAR problem does not change over time, and it needs to be solved just once the flows are established. Thus, the *network configuration* is stable under this assumption. The network configuration is defined as the set of selected paths for traffic flows and the link states. In contrast, in networks with time-varying flows, the instance of optimization problem evolves across time due to changes in traffic demands. In this case, we need to solve a sequence of ALR-based EAR problems repeatedly, and constantly reconsider the network configuration. At each iteration of optimization, there might be some differences between the optimal solution and the actual network configuration. Since the optimum configuration changes over time in response to the dynamically changing traffic demands, the network must be reconfigured at each iteration to have the minimum energy cost. The *network reconfiguration* includes the traffic flow reroutings and the link state transitions.

The network reconfigurations introduce instability to the system and decrease the QoS [5]. A link rate transition leads to packet loss and delay because it requires a considerable amount of time [6], and the link is not functional during the transition time. Also, flow rerouting affects the order of packets, and they can be received out of order. The number of reconfiguration needs to be minimized to avoid their negative impacts [5]. To the best of our knowledge, previous ALR-based EAR studies did not consider the effects of network reconfigurations in their path optimization.

This work aims to reduce the number of network changes in the periodic ALR-based EAR. Our approach is to select a network configuration in each time-slot which can achieve the following objectives: (i) it reduces the network energy consumption in the current time-slot, and (ii) it needs the minimum number of changes to adapt to the traffic demands in the future time-slots. Unlike existing ALR-based EAR approaches which considered the first objective, this approach does not focus only on the energy consumption in the current time-slot. It also considers the future changes in traffic demands and proactively plans to avoid significant network reconfigurations in future time-slots. Our approach stabilizes the network by

selecting a configuration which needs a few changes to be adapted to the future traffic demands.

A prediction algorithm is required in this approach to forecast the traffic demands in future time-slots and to eliminate the temporary reconfigurations. Any *multiple-step-ahead* time-series predictor can be exploited to achieve the predicted values. However, in order to fix a model, we employed our multiple-step-ahead prediction algorithm [7]. It models the traffic multiscale behaviour using a set of Gaussian Process Regression (GPR) learners. GPR is a kernel-based learning algorithm which can handle traffic characteristics such as short/long range dependency, self-similarity, and periodicity [8]. In [7], we showed that our multiple-step-ahead prediction algorithm outperforms other time-series predictors.

The contributions of this work are as follows:

- We design an innovative energy-efficient framework using the multiple-step-ahead prediction of the traffic load which is called Predictive ALR (PALR). It reduces the network energy consumption and avoids the QoS degradation.
- We formulate the optimization problem based on integer linear programming (ILP) to minimize both energy and reconfiguration costs.
- We advocate a heuristic algorithm based on simulated annealing (SA) which efficiently solves the ILP problem.

The remainder of this paper is organized as follows: Section II provides a summary of the existing work in ALR and traffic prediction. The ALR-based EAR is explained in Section III. Section IV describes the proposed approach. The experimental results are reported in Section V, and finally, the conclusion is drawn.

II. Related Work

ALR is an energy saving technique which establishes a relationship between the traffic workload and the power consumption. It consists of three broad classes according to its operation timescale [9]: (i) Demand-timescale rate adaptation (DTRA) with the period ranges from seconds to minutes, (ii) Packet-timescale rate adaptation (PTRA) which works in timescales of microseconds to milliseconds, and Bit-timescale rate adaptation (BTRA) that involves nanoseconds periods.

The BTRA and PTRA techniques apply to the individual network devices in the hardware-level and link-level solutions. They have been studied for real-time applications and different network protocols [10]. Also, the power-proportionality gained by embedding these techniques in various interfaces including small form-factor pluggable (SFP) modules and integrated twisted-pair Ethernet ports has been measured [9]. While the PTRA and BTRA techniques provide significant power saving [9], they cause performance degradation by triggering many state transitions [6].

The DTRA methods have been involved in the *network-level* solutions also known as energy-aware routing (EAR) algorithms applied to different networks such as SDN [11], and data center networks [12]. In EAR, the energy saving problem is solved as a multicommodity flow (MCF) optimization model considering the speed-power curve of the networking devices. The volumes of the traffic demands are used as inputs to the MCF optimization. Unfortunately, in the presence of variable traffic, the network re-optimization imposes many reconfigurations which are the consequences of the temporal flow allocations and link speed transitions. Our proposed algorithm integrates traffic prediction into the EAR at the demand-timescale (e.g., DTRA).

TABLE I
Model notation

Variable	Description
$\mathcal{L}$	the set of links in the network
$\mathcal{S}$	the set of possible states for a link
e_s	interface power consumption in state s
c_s	link capacity in state s
$u_l(t)$	utilization of link l at time t
$v_{l,s}(t)$	a binary variable which is equal to 1 if link l operates in state s at time t
$\mathcal{K}$	set of all the traffic flows in network
$d_k(t)$	bit-rate of flow k at time t
P_k	set of precalculated paths for flow k
$w_{p,k}(t)$	a binary variable which is equal to 1 if path p has been selected for flow k at time t
$E(t)$	network power consumption at time t
$R(t)$	number of reconfigurations from time $t-1$ to t
$O(t)$	Value of cost function at time t
I_t	available traffic information at time t
T	prediction horizon (the number of future steps of traffic)

Traffic forecasting can be used to avoid transient routing decisions. It computes future traffic demands for decision-making process. In [13] a PTRA method has been proposed which relies on traffic forecasting to estimate the number of packets that may arrive in the next time interval. In [14], a time window prediction (TWP) scheme has been proposed to reduce erroneous periods of sleep. Unlike prior prediction-based energy-saving proposals which employ forecasting in packet-timescale link-level solutions, in this work, we employed traffic prediction in demand-timescales on the network level. The prediction algorithm used in this paper has been proposed in [7]. It is based on Gaussian process regression (GPR) framework [8]. It exploits multiscale traffic behavior to reduce error propagation in multiple-step-ahead prediction. In [7], we showed this algorithm outperforms powerful time-series predictors.

III. Problem Definition

ALR has been proposed to reduce network power consumption. Network interfaces consume a notable portion of energy. Since the interface power consumption is independent of its utilization [6], ALR adjusts the interface capacity according to the link utilization to minimize the required energy. Generally, interfaces consume a lower amount of energy while they operate with lower capacities. There are ALR schemes with two-state interfaces. For examples, in Energy Efficient Ethernet (EEE) which has been introduced in standard IEEE 802.3az, interfaces support two states: a normal mode, and a

TABLE II
LINK POWER CONSUMPTION IN DIFFERENT STATES [3]

State s	Rate c_s	Power Consumption [mW] e_s
1	100Mbps	351
2	1Gbps	697
3	10Gbps	2600

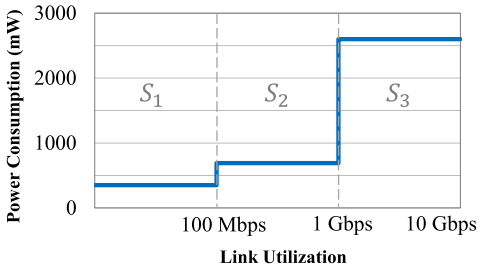


Fig. 1. Discrete step function of link power consumption $E(f_l)$

low power idle (LPI) mode [3]. In a general ALR schema, interfaces operate in S states (Table I presents the notation). For example, the three-state configuration illustrated in Table II has been derived from technical documents of Ethernet devices [3]. The discrete step increasing function $E(u_l(t))$ in Figure 1 gives the power consumption of ALR-enabled link l with load $u_l(t)$ based on Table II. Power consumption decreases with transmission rate (i.e. $e_{s-1} < e_s \Leftrightarrow c_{s-1} < c_s$).

The goal is to minimize the overall power consumption while preserving adequate link capacity for the flows. There are K flows in the network and a set of precalculated paths for each flow. In the case of constant demands, the ALR-based EAR is solved when the flows are established. The optimization at time t_0 can be formulated as:

$$\text{minimize} \quad \sum_{l\in\mathcal{L}}\sum_{s\in\mathcal{S}} e_s v_{l,s}(t_0) \tag{1}$$

subject to :

$$\sum_{p\in P_k} w_{p,k}(t_0) = 1 \qquad \forall k \in \mathcal{K} \tag{2}$$

$$\sum_{s\in\mathcal{S}} v_{l,s}(t_0) = 1 \qquad \forall l \in \mathcal{L} \tag{3}$$

$$\sum_{s\in\mathcal{S}} (v_{l,s}(t_0)\cdot c_s) \geq u_l(t_0) \qquad \forall l \in \mathcal{L} \tag{4}$$

$w_{p,k}(t_0) \in \{0,1\}$ equals 1 if path $p \in P_k$ is selected for flow k at time t_0. Variable $v_{l,s}(t_0) \in \{0,1\}$ equals 1 if link l is in state s at t_0. Objective function (1) considers only the energy consumption. Constraint (2) guarantees that there is no unallocated flow. Equation (3) means link l operates only in one state during time-slot t_0. Equation (4) ensures the link capacity is more than the load on the link. The traffic load on link l at t is the sum of traffic demands that are routed on l:

$$u_l(t) = \sum_{k\in\mathcal{K}}\sum_{\forall p: l\in p} w_{p,k}(t)\cdot d_k(t) \qquad \forall l \in \mathcal{L}. \tag{5}$$

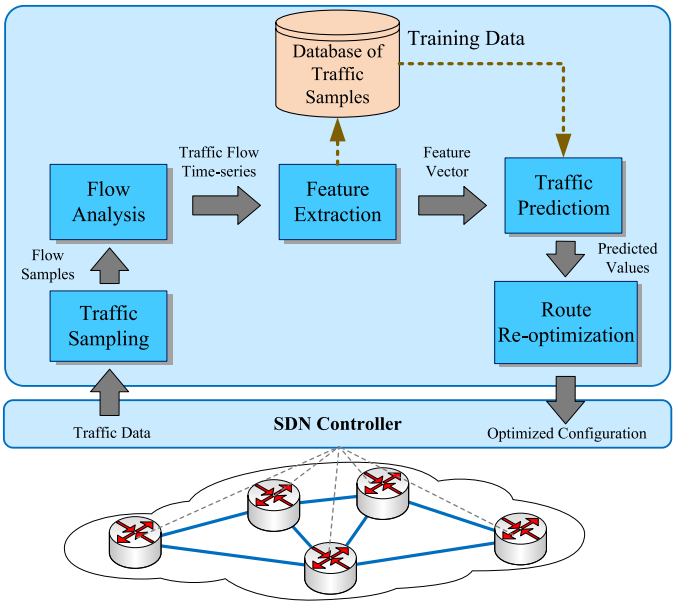


Fig. 2. The proposed Framework for PALR

There are situations when demands are not constant, and $d_k(t)$ takes different values in successive intervals. In such cases, the routing configurations (i.e. $w_{p,k}(t)$ and $v_{l,s}(t)$) must be adjusted in every time-slot to maintain the optimal power consumption. These network re-optimizations provoke a vast number of changes (i.e., flow rerouting, and link state transitions) which significantly reduce the network performance. Therefore, the ALR-based EAR needs to compromise between two targets in each re-optimization. On the one hand, the network must be energy efficient according to the current demands. On the other hand, the number of modifications to the configuration must be minimized. In this work, we propose an effective approach to achieve this trade-off. In our approach, the selected configuration minimizes the power consumption in current time-slot *while* it needs the least number of changes to become adapted to the demands in future time-slots.

IV. PREDICTIVE ALR

This section explains our proposed Predictive ALR (PALR) framework shown in Figure 2. Traffic is monitored using the SDN controller. After performing flow analysis on the observed traffic data, feature vectors are extracted and stored in a database. The historical data in the database is used for initializing and training the predictor which estimates the future steps of traffic given a new sample. The outcome of the predictor is used in the optimization to find the optimal paths for flows. The elements of the framework are explained in this section.

A. Traffic Sampling and Flow Analysis

The traffic sampling and the flow analysis modules monitor and collect the traffic samples for each flow. The definition of traffic flow depends on the type of network. For example, in an IP network, all the packets that have the same protocol type,

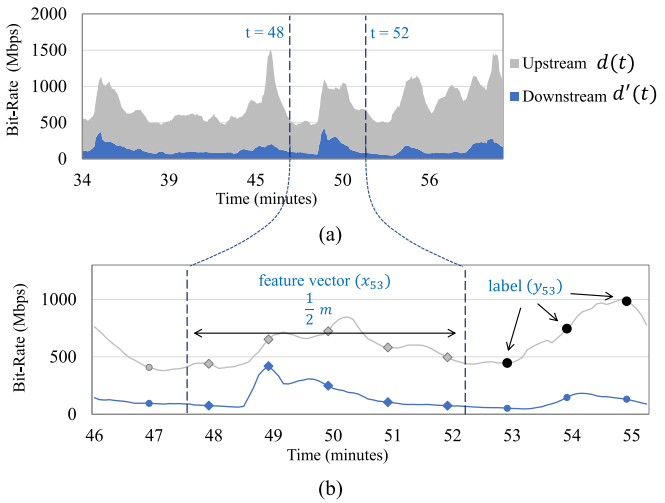


Fig. 3. Traffic Sampling and Feature Extraction

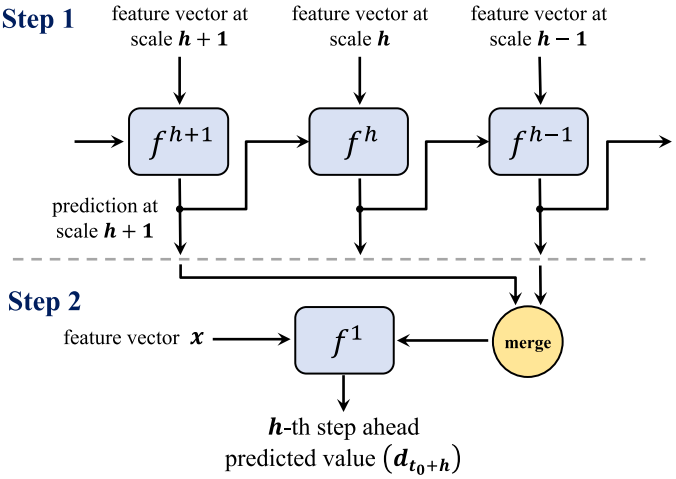


Fig. 4. Multiple-step-ahead traffic prediction algorithm [7]

source/destination IP, and source/destination port belong to the same traffic flow. In the networks where there is a massive number of traffic flows, it is not applicable to track and monitor all the flows. Instead, to reduce the monitoring and prediction overhead, it is possible to observe only the big flows. It is well-known that a large portion of traffic is carried by a small number of flows called big flows (or elephant flows). In [15], prediction of elephant flows has been employed to manage inter-data-center traffic. The energy consumption can be optimized by managing the big flows inside the network while the small flows (or mice flows) can be routed using a general-purpose routing algorithm. So, there is no need to monitor and track small flows. The flow analysis has to provide the information of upstream and downstream traffic flows.

B. Feature Extraction and Traffic Prediction

The goal of this part is to train a prediction algorithm with horizon T. Prediction horizon (T) is defined as the number of future time-slots that are predicted by the algorithm. First, we need to extract the features of traffic samples and create a training set. Consider $\bar{d}(t) = (d(t), d^{'}(t))$ as the bandwidth utilization of flow at t where $d(t)$ and $d^{'}(t)$ are upstream and downstream values respectively. A traffic sample (x_t, y_t) is composed of feature vector x_t, and multivariate label y_t. The length of the feature vector is m while $m/2$ samples are taken from upstream and $m/2$ samples are from downstream:

$$\begin{aligned} x_t \;=\; & [d(t-1),\ d(t-2),\ \ldots,\ d(t-m/2), \\ & d^{'}(t-1),\ d^{'}(t-2),\ \ldots,\ d^{'}(t-m/2)]. \end{aligned} \quad (6)$$

Since there is a relation between upstream and downstream demand, both are used in the feature vector to increase the prediction accuracy. The length of multivariate label y_t is equal to the prediction horizon T. It includes future values of flows:

$$y_t = [d(t), d(t+1), ..., d(t+T-1)] \quad (7)$$

Figure 3 illustrates the feature extraction for upstream traffic at $t = 53$. The bit-rates of $d(t)$ and $d^{'}(t)$ are shown in Figure 3.a. Figure 3.b presents traffic sampling at the time-scale of 1 minute. Length of feature vector x_{53} is 10 while 5 samples are from $d(t)$, and 5 samples are from $d^{'}(t)$. Length of multivariate label $y_{53} = [d(53), d(54), d(55)]$ is equal to $T = 3$. The samples that are generated in this examples are utilized for 3-step-ahead prediction of upstream demand. The samples for downstream prediction have the same feature vector. Unless their labels include the future steps of $d^{'}(t)$.

Database $\mathcal{D} = \{q_{t_i} \mid i = 1, 2, ..., N\}$ includes N historical traffic samples from different flows. $\mathcal{D}$ is used to train a supervised machine learning algorithm f. At time t, predictor f uses x_t as input to predict $y_t^{'}$:

$$f(x_t) = y_t^{'} \quad (8)$$

$$y_t^{'} = [d(t), d(t+1), ..., d(t+T-1)] + \varepsilon_t \quad (9)$$

where $y_t^{'}$ is the estimation for y_t, and ε_t is the prediction error (a vector with length T). The proposed framework for the ALR-based EAR (in Figure 2) is not restricted to a particular prediction algorithm, and it is possible to employ any predictor including ARIMA, FARIMA, LSTM, etc.

We employed our multiple-step-ahead traffic predictor proposed in [7]. Unlike other time-series predictors, it has been designed based on traffic characteristics. The key idea behind this algorithm is the multiscale behavior of traffic. It employed traffic information from different time-scales to reduce error propagation. The workflow model of our algorithm is presented in Figure 4. It consists of two steps. There are H GPR experts in Step 1 which predict traffic at different time-scales. In Step 2, the outcomes of experts f^{h+1} and f^{h-1} are merged and used to predict h-th step ahead value of traffic ($d(t_0+h)$).

C. Route Re-optimization

The optimization module selects the optimal routes for the traffic flows to reduce network energy consumption and to minimize future reconfigurations in the network according to

the predicted bandwidth of flows. Network power consumption at t is computed as:

$$E(t) = \sum_{l \in \mathcal{L}} \sum_{s \in \mathcal{S}} e_s v_{l,s}(t) \tag{10}$$

The number of reconfigurations in the network at time t is:

$$\begin{aligned} R(t) \quad = \quad & \frac{1}{2} \sum_{l \in \mathcal{L}} \sum_{s \in \mathcal{S}} |v_{l,s}(t) - v_{l,s}(t-1)| \\ + \quad & \frac{1}{2} \sum_{k \in \mathcal{K}} \sum_{p \in P_k} |w_{p,k}(t) - w_{p,k}(t-1)| \,. \end{aligned} \tag{11}$$

$R(t)$ is the sum of the number of link state transitions and the number of flow reroutings. The goal is to find a network configuration that minimizes the power consumption (in current time-slot) and needs the least number of modifications to stay optimal (in future time-slots). O_t is the cost function at t:

$$O(t) = \frac{E(t)}{E_{max}} + \frac{R(t)}{R_{max}}. \tag{12}$$

R_{max} is the maximum number of changes in one iteration:

$$R_{max} = K + L. \tag{13}$$

It means there are K flows that can be rerouted and L links that their rate can be switched. The maximum consumption (E_{max}) is achieved when all the interfaces are working with maximum capacity. $O(t)$ is the sum of two normalized terms, and it takes values in the range $[0, 2]$.

Network is re-optimized at the end of each time-slot. At t_0, the optimization model considers the traffic data in the range of time-slots $\mathcal{T} = \{t_0 + i \mid i = 0, ..., T\}$. The current volumes of demands ($\{d_k(t_0) \mid k \in \mathcal{K}\}$) are known, and predictor f gives the volumes of demands during the next T steps ($\{d_k(t_0+i) \mid k \in \mathcal{K}, i = 1, ..., T\}$). Our proposed optimization model is:

$$\text{minimize} \quad \sum_{t \in \mathcal{T}} O(t) \tag{14}$$

subject to :

$$\sum_{p \in P_k} w_{p,k}(t) = 1 \qquad \forall t \in \mathcal{T},\ \forall k \in \mathcal{K} \tag{15}$$

$$\sum_{s \in \mathcal{S}} v_{l,s}(t) = 1 \qquad \forall t \in \mathcal{T},\ \forall l \in \mathcal{L} \tag{16}$$

$$\sum_{s \in \mathcal{S}} (v_{l,s}(t) \cdot c_s) \geq u_l(t) \qquad \forall t \in \mathcal{T},\ \forall l \in \mathcal{L} \tag{17}$$

The model discussed in Section III (i.e., ALR-based EAR without prediction) and the proposed model (i.e., ALR-based EAR using prediction) are mainly different in their objective functions. The former focused on the consumption in current time-slot while the latter considers consumption and number of changes during current and future steps. Both models are NP-hard problems. The computational requirements of the proposed model increase exponentially by the rise in the number of precalculated paths, the number of traffic flows, and length of prediction horizon T. There are different approaches to resolve such an NP-hard problem. For example, the ALR-based EAR (without prediction) has been solved using a greedy algorithm in [4]. In this work, we designed a heuristic algorithm based on simulated annealing (SA) that efficiently solves the problem.

Algorithm 1 Simulated Annealing

Input: A_{min}, A_{max}, A_d, I_{max}
Output: V^*, W^*

$A = A_{max},\ V = \bar{V},\ W = \bar{W},\ O^* = \infty,\ i = 1$
$O = Cost(V, W)$
while $A \geq A_{min}$ **do**
  if $Valid(V, W)$ and $O < O^*$ **then**
    $V^* = V,\quad W^* = W,\quad O^* = O$
  end if
  $(V^{'}, W') = GenerateNeighborSolution(V, W)$
  $O^{'} = Cost(V^{'}, W^{'})$
  $r = GenerateRandomNumber()$
  $i = i + 1$
  if $Pr(O, O^{'}, A) \geq r$ **then**
    $V = V^{'},\quad W = W^{'},\quad O = O^{'}$
  end if
  if $i \geq I_{max}$ **then**
    $A = A_d \cdot A,\quad i = 1$
  end if
end while
return V^*, W^*

Simulated annealing [16] has been motivated by the physical process of heating a material and then slowly lowering the temperature to minimize the system energy. It is a popular algorithm for finding the global optimum in a large search space with many local optimums. SA accepts solutions that are worse than the current solution with some probability to escape local optimums during the search process. The probability of accepting a worse solution lowers with a control parameter called *temperature*. SA explores the search space when the temperature is high, and it converges as the temperature is decreasing. Also, the probability of accepting a worse solution has a direct relationship with the difference between the objectives of the current and new solution. SA needs four parameters: starting temperature (A_{max}), ending temperature (A_{min}), temperature decrement (A_d), and the number of iterations at each temperature (I_{max}). Generally, A_{max} is set to 1, and A_{min} is set to a value close to zero (e.g., 0.000001).

Algorithm 1 presents the proposed SA method. Temperature is initially set to A_{max} and decreased to reach A_{min}. SA starts from an initial solution ($\bar{V}$ and $\bar{W}$) and generates a random neighbor ($V^{'}$ and $W^{'}$) of the current solution (V and W) in each iteration. The generated solution is accepted (and becomes the current solution) with probability $Pr(O, O^{'}, A)$:

$$Pr(O, O^{'}, A) = exp\left(-\frac{O^{'} - O}{A}\right) \tag{18}$$

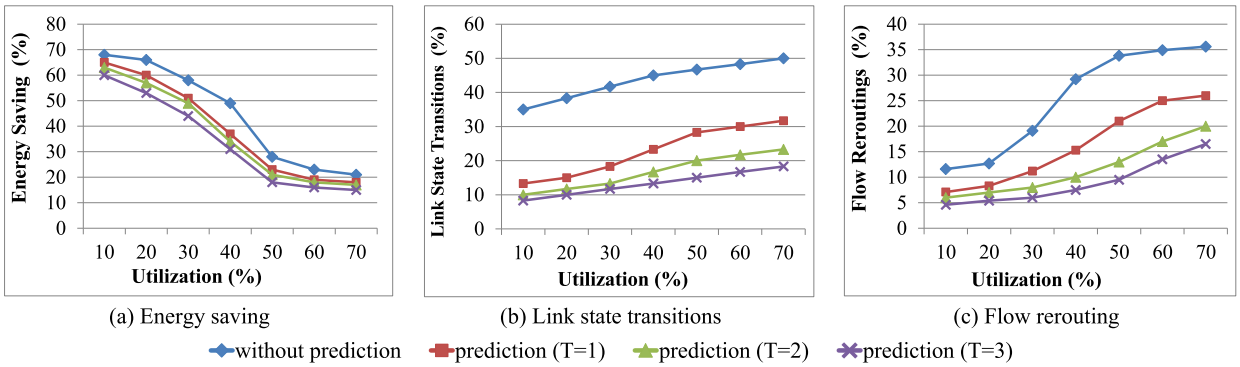


Fig. 5. Comparison of traditional model and the proposed model with different prediction horizons

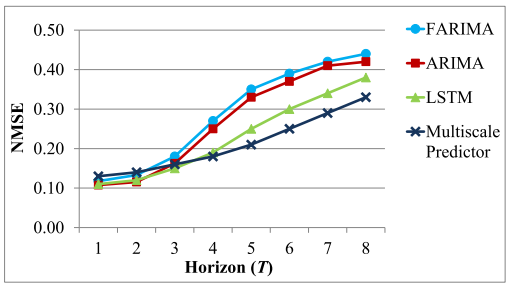


Fig. 6. multiple-steps-ahead prediction on the Abilene network traffic data at time-scale of 15 minutes

where A is the current temperature and O and $O^{'}$ are respectively the costs of current and generated solutions. Note that the new solution is accepted if it has a lower cost compared to the current solution (because $Pr(O, O^{'}, A)$ takes a value more than 1). In algorithm 1, V and W determine the configuration:

$$V = \{v_{l,s}(t) \mid \forall s \in \mathcal{S},\ \forall l \in \mathcal{L},\ \forall t \in \mathcal{T}\}, \tag{19}$$

$$W = \{w_{p,k}(t) \mid \forall k \in \mathcal{K},\ \forall p \in P_k,\ \forall t \in \mathcal{T}\}. \tag{20}$$

The initial point ($\bar{V}$ and $\bar{W}$) is a solution in which the current configuration at $t_0 - 1$ remains unchanged during $\mathcal{T}$. $Cost(V, W)$ is calculated using Equation (12). Boolean function $Valid(V, W)$ returns $True$ if V and W satisfy Equations (15), (16), and (17). $GenerateRandomNumber$ returns a random value between 0 and 1. Function $GenerateNeighborSolution$ gives a random neighbor solution. Starting from current solution, it selects a random flow k and a random path in P_k to route k on P_k at a random time-slot $t^{'} \in \mathcal{T}$. The new route for k remains unchanged in the next time-slots t (when $t \geq t^{'}$). Then, it randomly changes the link states in the previous and new routes considering constraints (16) and (17) at time-slots $t \geq t^{'}$.

V. Experimental Results

A. Setup

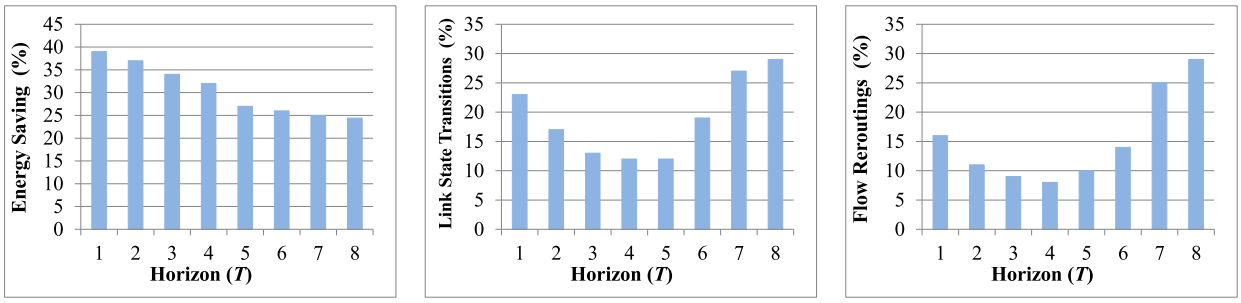
We used the topology and the traffic traces from the Abilene network in our experiments. Our tests have been done at the timescale of 15 minutes. Hence, the network configuration is updated every 15 minutes. An instance of the multiple-step-ahead predictor has been used. It has been trained only once at the beginning with $m = 10$ using 3000 samples. The prediction is made for each flow separately. Energy consumption of network links are given in Table II. We emulated 500 traffic flows based on the data of real traffic flows collected from Abilene nodes during 2007-01-01 and 2007-10-14 [17]. There are at most 3 precalculated paths (3 first shortest) between each pair of nodes in the network. We performed the repeated network optimization using two models: ALR-based EAR with/without prediction. The optimization problems have been solved using SA in Algorithm 1. In each iteration, the number of link state switches, the number of reroutings, and the total energy consumption are measured.

B. Results

The prediction error of our multiple-step-ahead prediction algorithm (i.e., Multiscale Predictor) has been shown in Figure 6. The prediction error is measured using Normalized Mean Squared Error (NMSE) [7]. The model has been compared to 3 well-known time-series predictors (ARIMA, FARIMA, and LSTM). Fig. 6 includes 8 steps ahead prediction results ($T = 8$) while the time-scale is 5 minutes. The prediction error of different predictors increases with T. In the first steps, the algorithms have almost the same performance. The propagation of error in our algorithm (i.e., the Multiscale Predictor) is less than other predictors.

Figure 5 illustrates the comparison of network optimization using different models. The horizontal axis shows the *average link utilization*. Different levels of utilization are achieved by increasing the bit-rate of traffic flows. In Figure 5.a, the results of energy saving for different approaches (compared to the maximum consumption) are shown. The network consumes the maximum power when there is no ALR and the interfaces operate with maximum capacity. Figures 5.b and 5.c respectively show the percentage of interfaces that need state transition and the percentage of flows that must be rerouted in each iteration.

According to Figure 5.a, the ALR-based EAR without prediction saves 70% of energy consumption when the average link utilization is 10%. It has better performance for energy



(a) Energy saving (b) Link state transitions (c) Flow rerouting

Fig. 7. Performance of the proposed model using different prediction horizons

saving compared to the proposed approach. However, it causes a large number of network modifications. For example, according to Figures 5.b and 5.c, when the average link utilization is 10%, it changes the state of 35% of interfaces and reroutes 12% traffic flows in each iteration (i.e., every 15 minutes) which lead to significant QoS degradation. As shown, our proposed algorithm (ALR-based EAR with prediction, $T = 1$) saves 60% of energy which is less than traditional algorithm (ALR-based ERA without prediction). However, it changes only 9% of the link states and around 5% of the routes. This means the ALR-based EAR using prediction provides a trade-off between energy saving and network changes. This results are almost the same for ALR-based EAR with longer prediction horizons ($T > 1$). As the T increases, the energy saving drops slightly. At the same time, the number of changes to the network is reduced significantly.

Figure 7 shows the performance of LRA-based using prediction with different prediction horizons. The percentage of network reconfigurations is decreased (according to Figures 5.b and 5.c) from $T = 1$ to $T = 5$. However, as the prediction error raises, the performance of the model (for reducing the number of changes) decreases. So, the number of reconfigurations increases when $T \geq 6$. Therefore, T is limited by the level of prediction error.

VI. Conclusion

We provided an optimization framework for networks with fluctuating traffic demands using ALR-based EAR. Traffic prediction has been involved in the proposed approach to proactively control the number of changes and preserve QoS during network re-optimizations. The optimization problem has been formulated as a ILP model and solved using a SA algorithm. Our results confirm the proposed approach significantly reduces the number of changes while it provides energy conservation.

Acknowledgment

The authors thank NSERC and Ciena for funding the project CRDPJ 461084. This research also receives support from the Canada Research Chair, Tier 1, hold by Mohamed Cheriet.

References

[1] G.-R. Liu, P. Lin, and M. K. Awad, "Modeling energy saving mechanism for green routers," *IEEE Transactions on Green Communications and Networking*, 2018.

[2] J. A. Manjate, M. Hidell, and P. Sjödin, "Can energy-aware routing improve the energy savings of energy-efficient ethernet?" *IEEE Transactions on Green Communications and Networking*, 2018.

[3] S. Vitturi and F. Tramarin, "Energy efficient ethernet for real-time industrial networks," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 228–237, 2015.

[4] J. Tang, B. Mumey, Y. Xing, and A. Johnson, "On exploiting flow allocation with rate adaptation for green networking," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1683–1691.

[5] A. Destounis, S. Paris, L. Maggi, G. S. Paschos, and J. Leguay, "Minimum cost sdn routing with reconfiguration frequency constraints," *IEEE/ACM Trans. on Networking*, vol. 26, no. 4, pp. 1577–1590, 2018.

[6] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (alr)," *IEEE Transactions on Computers*, vol. 57, no. 4, pp. 448–461, April 2008.

[7] A. Bayati, K. Nguyen, and M. Cheriet, "Multiple-step-ahead traffic prediction in high-speed networks," *IEEE Communications Letters*, pp. 1–1, 2018.

[8] A. Bayati, V. Asghari, K. K. Nguyen, and M. Cheriet, "Gaussian process regression based traffic modeling and prediction in High-Speed networks," in *2016 IEEE Global Communications Conf. (Globecom2016 CQRM)*, Washington, USA, Dec. 2016.

[9] A. Francini, S. Fortune, T. Klein, and M. Ricca, "A low-cost methodology for profiling the power consumption of network equipment," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 250–256, May 2015.

[10] A. Francini, "Selection of a rate adaptation scheme for network hardware," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 2831–2835.

[11] M. Rahnamay-Naeini, S. S. Baidya, E. Siavashi, and N. Ghani, "A traffic and resource-aware energy-saving mechanism in software defined networks," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2016, pp. 1–5.

[12] L. Wang, F. Zhang, C. Hou, J. A. Aroca, and Z. Liu, "Incorporating rate adaptation into green networking for future data centers," in *Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on*, Aug 2013, pp. 106–109.

[13] M. Gupta and S. Singh, "Using low-power modes for energy conservation in ethernet lans," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, 2007, pp. 2451–2455.

[14] G. Ananthanarayanan and R. H. Katz, "Greening the switch," in *Proceedings of the 2008 Conf. on Power Aware Computing and Systems*, ser. HotPower'08, USA, 2008, pp. 7–7.

[15] Y. Li, H. Liu, W. Yang, D. Hu, X. Wang, and W. Xu, "Predicting inter-data-center network traffic using elephant flow and sublink information," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 782–792, Dec 2016.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 83–86, Jan. 2006.