Bandwidth Scheduling for Energy Efficiency in High-performance Networks

Tong Shu, *Student Member, IEEE,* and Chase Q. Wu, *Member, IEEE*

***Abstract*—The transfer of big data in various applications across high-performance networks (HPNs) in a national or international scope consumes a significant amount of energy on a daily basis. However, most existing bandwidth scheduling algorithms only consider traditional objectives such as data transfer time minimization, and very limited efforts have been devoted to energy efficiency in HPNs. In this paper, we consider two widely adopted power models, i.e. power-down and speed-scaling, and formulate two instant bandwidth scheduling problems to minimize energy consumption under data transfer deadline and reliability constraints. We prove the NP-completeness of both problems, and design a fully polynomial time approximation scheme (FPTAS) for the problem using the power-down model. We also design an approximation algorithm and a heuristic approach that considers the tradeoff between objective optimality and time cost in practice for the problem using the speed-scaling model. The performance superiority of the proposed solutions is illustrated by extensive results based on both simulated and real-life networks in comparison with existing methods.**

***Index Terms*—High-performance networks, bandwidth scheduling, energy efficiency.**

I. Introduction

MANY large-scale applications in various science, engineering and business domains are generating colossal amounts of data, now frequently termed as "big data", which are carried by high-performance networks (HPNs) with the capability of bandwidth provisioning to support various remote operations. Several HPN projects are currently underway, including On-demand Secure Circuits and Advance Reservation System (OSCARS) [1] of the Energy Sciences network (ESnet) and Interoperable On-demand Network (ION) of Internet2 [2].

The trunk links in HPNs are oftentimes underutilized as exemplified by ESnet [33]. However, due to a dynamic distribution of traffic loads, network devices are always powered on to support long-duration big data transfer, which consumes a significant amount of energy on a daily basis. Hence, it is particularly important to achieve energy-proportional networking in HPNs. The energy consumption of network devices consists of two components, i.e. static energy consumption (SEC) and dynamic energy consumption (DEC). In many cases, SEC could take up around 90% of the total consumed energy. Green networking is mainly based on two types of techniques, i.e. traffic consolidation to reduce SEC by powering down idle network devices, and load balancing to reduce DEC using adaptive link rate supported by dynamic voltage and frequency scaling (DVFS) for speed-scaling power consumption, or a combination of both. Researchers have applied such techniques to different network environments including the Internet, which is an extreme-scale network without a centralized scheduler, and data center networks (DCNs), which are small-scale local access networks with a hierarchical topology, and proposed various network-layer solutions. However, the efforts made in HPNs along these lines are still quite limited.

Unlike the Internet and DCNs, HPNs are wide-area backbone networks with an irregular topology to provision dedicated channels through bandwidth reservation for big data transfer. Moreover, bandwidth provisioning fits naturally in the framework of the emerging Software-Defined Networking (SDN) technology, where the controller is separated from the data plane and maintains the global network topology. In general, the HPN infrastructures such as edge devices, core switches, and backbone routers are coordinated by a management framework, namely control plane, which is responsible for reserving link bandwidths, setting up end-to-end network paths, and releasing resources when tasks are completed. As the central unit of a generalized control plane, the bandwidth scheduler computes appropriate network paths and allocates link bandwidths to meet specific user requests based on network topology and bandwidth availability [24]. Different from existing bandwidth scheduling problems that mainly consider traditional optimization objectives such as minimizing the time span of a data transfer, our work focuses on the energy efficiency of bandwidth scheduling in HPNs.

We take two orthogonal approaches to achieve energy efficiency in HPNs: i) reduce SEC, the main energy cost, based on the power-down model, taking transitional power consumption and time into consideration while ignoring the non-linear variation of dynamic power consumption (DPC); ii) reduce DEC based on the speed-scaling model since frequently shutting down network devices may disrupt network connectivity and undermine the life span of affected devices. The speed-scaling model in existing research is divided into two categories: experimental power model based on the linear regression of device measurements, and theoretical power model including step functions and power functions based on the property of a single gate circuit. The former does not consider the non-linear variation of DPC, while the latter may lack accuracy in modeling the actual power consumption of network devices. To improve the modeling accuracy with a better alignment with practical measurements, we develop and adopt a DPC model based on polynomial functions.

Some preliminary results in this manuscript were published in a conference paper [28].

T. Shu and C.Q. Wu are with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA. Email: {ts372, chase.wu}@njit.edu. Please send all correspondence to Wu.

In this paper, based on the widely adopted power-down and speed-scaling models, we formulate two instant bandwidth scheduling problems to minimize energy consumption under data transfer deadline and reliability constraints. We prove the NP-completeness of both problems and design a class of approximation and heuristic algorithms, referred to as Smart Advance reserVation for Energy Efficiency (SAVEE), taking into consideration energy consumption, reliability, and transfer time in practice. The performance superiority of the proposed solutions is illustrated by extensive results based on both simulated and real-life networks in comparison with existing methods.

The rest of the paper is organized as follows. Section II provides a survey of related work. We formulate two bandwidth scheduling problems in Section III and prove their NP-completeness in Section IV. In Section V, we design a fully polynomial time approximation scheme (FPTAS) for the problem using the power-down model. In Section VI, we design an ϵ-approximation algorithm and a heuristic for the problem using the speed-scaling model. Sections VII and VIII present the performance evaluation results.

II. Related Work

Since Gupta *et al.* first discussed energy saving by putting network components to sleep [20], many research efforts have been made in green networking. Existing techniques at the software level mainly fall in two categories: resource consolidation and energy-proportional computing [9].

A. Resource Consolidation

Resource consolidation, i.e. power-down, reduces energy waste due to the over-provisioning and over-dimensioning of network infrastructures [25], by turning off some lightly loaded routers and rerouting the network traffic on a selected set of active network equipments [26].

Some research on power-down policies is focused on energy-efficient routing on the Internet (no centralized scheduler) to address various issues, such as topology construction for full connectivity using the least number of network devices [13], routing stability during topology reconstruction, and compatibility with existing routing protocols [23].

Other efforts concentrate on centrally controlled networks with MPLS and RSVP-TE protocols. Zhang *et al.* proposed an intra-domain traffic engineering mechanism called GreenTE, a heuristic that deals with multiple traffic demands to maximize power saving on idle links and line cards under link utilization and packet delay constraints in commercial networks, such as AT&T [38]. In frame-based telecommunication infrastructures, Andrews *et al.* studied periodic scheduling and routing to minimize an active period per element within each frame, and designed a schedule with close-to-minimum delay in a linear topology. They extended it to an arbitrary topology by network partition, and provided a $O(\log \log N)$-approximation for power and delay minimization by reducing switching via routing [5]. Addis *et al.* formulated multiperiod energy-aware fixed and variable routing as an ILP problem for multiple traffic demands with the constraints on the chassis capacity and interperiod line-card reliability restriction among a set of traffic scenarios and proposed a heuristic online approach [3]. Since ILP cannot be solved in polynomial time, their solutions were only verified in a small-scale network of 9 nodes. Zhang *et al.* investigated an NP-complete problem of utilizing correlated traffic demands to conduct power-aware configuration of points of presence (PoPs) [39]. They designed an optimal algorithm when each chassis has 2 ports, and a $P/2$-approximation and a $2 \ln N$-approximation algorithm for the general problem, where P is the number of ports per chassis and N is the number of chassis. However, their work was focused on static network construction, not dynamic routing or scheduling.

There also exists some work on energy efficiency in data center networks (DCNs). Considering the fact that bandwidth demands of different flows do not peak simultaneously, Wang *et al.* formulated the integration of link rate adaptation and traffic consolidation based on correlation analysis among flows in DCNs as an NP-complete flow assignment problem for minimum power consumption of switches, and proposed a correlation-aware heuristic to select a minimum subset of devices [37]. Li *et al.* proposed an energy-aware instant flow scheduling solution in SDNs for DCNs with homogeneous switches through an exclusive routing scheme for each flow [22]. Note that DCNs are local access networks with a hierarchical organization of switches to interconnect tightly-coupled high-performance servers, while HPNs considered in our work are one type of backbone networks with irregular wide-area footprints to support dedicated data transfer.

While power-down policies save significant SEC, frequently switching on/off devices involves considerable time and energy overhead. However, the above work does not consider booting-up and shutting-down time (several minutes) that is critical in HPNs, and is not intended for HPNs where a single big data transfer may consume substantial energy. Our work fills in the blank for energy efficiency in HPNs.

B. Energy-proportional Computing

Energy-proportional computing, i.e. speed scaling, ensures that power consumption scales proportionally with resource utilization. Typical examples include dynamic voltage and frequency scaling (DVFS) and adaptive link rate (ALR) [8]. Energy saving from speed scaling largely depends on specific power models of devices.

Chabarek *et al.* and Vishwanath *et al.* measured the power consumption of routers/switches equipped with different line cards under various traffic conditions through experiments, and developed power models at the granularity level of per chassis, line card or port [11], and at the granularity of per-packet processing and per-byte store-and-forward packet handling operations [35], respectively. Since measurement-based power models are generated by linear regression, they are mainly used in system design for simplicity.

Dynamic power models are sometimes considered as step functions with respect to the data rate. Tang *et al.* formulated problems of single-session and multi-session flow allocation with rate adaptation (SF-RAP and MF-RAP) for multi-path transfer, to find a flow allocation on given candidate paths for each session to minimize incremental power consumption. They proved the NP-hardness of both problems and presented

a 2-approximation for SF-RAP and an LP-based heuristic for MF-RAP [29]. According to the advanced configuration and power interface specification that introduces low power idle (LPI) and ALR, Bolla *et al.* proposed a queuing theory-based model to describe the impact of these technologies on the packet processing engine with multiple parallel pipelines inside a single line card, and formulated a problem to capture the tradeoff between power consumption and packet latency [10].

Other research considers dynamic power models as power functions of data rate r, i.e. $P(r) = \begin{cases} 0, r = 0 \\ \sigma + \mu r^{\alpha}, r > 0 \end{cases}$. i) Some work only considers DPC, i.e. speed-power curves ($\alpha > 1$) with $\sigma = 0$. Zhao *et al.* introduced a Nash bargaining framework that treats load balance and energy efficiency objectives as two virtual players in a game theoretic model [40]. Andrews *et al.* proposed Batch, SlowStart and queue-based rate adaptation policies for a single server to ensure various bounds on queue size (i.e. network stability) and energy usage, and showed that in a temporary sessions model, SlowStart with farthest-to-go/nearest-to-source scheduling provides the bounds on queue size and energy usage, and in a permanent sessions model, any above rate adaptation policy with weighted fair queueing scheduling provides a delay bound with respect to rates [6]. ii) Some other work considers both static ($\sigma > 0$) and DPC. Andrews *et al.* formulated a min-power multicommodity integral unicast routing problem that guarantees the bandwidth demand of each flow in an undirected graph, and proved that there is no bounded approximation. They provided a $O(\log^3 N)$-approximation for subadditive speed-power curves, where N is the network size, and used randomized rounding to provide a $O(\log^{\alpha-1} D)$-approximation for a superadditive speed-power curve $\mu_e r^{\alpha}$ per link e, where D is the maximum bandwidth demand of flows, and a $O(n + \log^{\alpha-1} D)$-approximation, where n is the number of flows, and a $O((1+\max_e\{\sigma_e/\mu_e\}^{1/\alpha})\log^{\alpha-1} D)$-approximation for speed-power curves $\sigma_e + \mu_e r^{\alpha}$ ($\alpha > 1$) [4]. Antoniadis *et al.* designed a $O(\log^{\alpha} n)$-approximation offline algorithm and a $O(\log^{3\alpha+1} n(\log\log n)^{2\alpha})$-competitive online algorithm for a unicast energy-efficient multicommodity edge routing problem of unsplittable unit flows in an undirected multi-graph with $\alpha > 1$ [7]. Wang *et al.* studied a scheduling and integral routing problem that considers deadline-constrained flows with given data sizes in DCNs, and proposed a P-solution for a special case with given routes [36]. They proved that the joint scheduling and routing problem is strongly NP-hard, and provided a $O(\lambda^{\alpha}(n^2 \log d)^{\alpha-1})$-approximation, where d is the maximum density of flows and $\lambda = (t_K - t_0)/\min_k(t_k - t_{k-1})$, where $t_0 \leq t_1 \leq \ldots \leq t_K$ are release times and deadlines of flows. Gupta *et al.* designed an α^{α}-competitive online greedy algorithm for a unicast energy efficient edge routing problem with splittable flows [19]. Since speed scaling is applied to routers more than links, Krishnaswamy *et al.* proposed a $O(\log^{12\alpha+5} N)$-approximation for a unicast energy-efficient vertex routing problem with splittable unit flows in an undirected multi-graph with $\alpha > 1$ [21].

In fact, the dynamics in the power consumption of routers, switches, and line cards in real networks are much more complex than the above power models. Fig. 1 shows the actual power consumption measurements of different Cisco routers in various settings. The simplified power models in the literature such as a step or power function are generally insufficient to model such complexities in real networks, such as Fig. 2 in [8]. In this paper, we develop a generalized power consumption model of a polynomial function for network devices and employ the speed-scaling technique based on this model to achieve energy efficiency in HPNs.

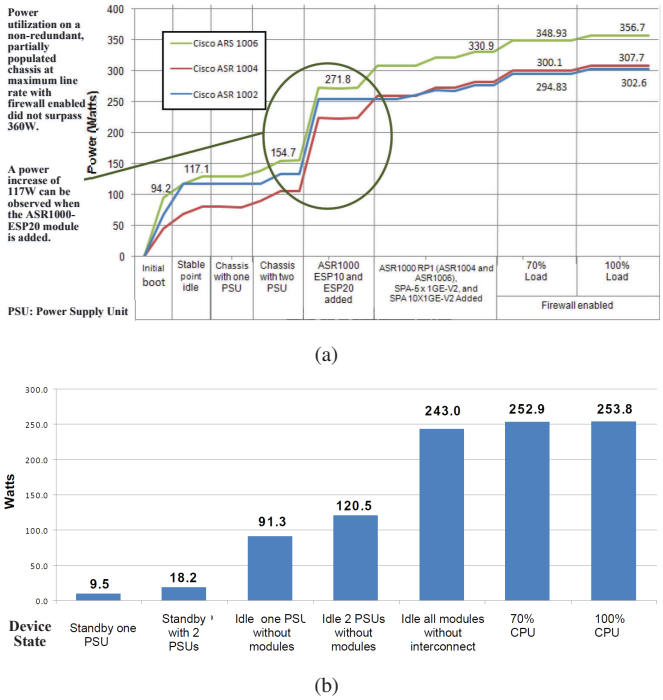


Fig. 1. Power profiles of Cisco routers: (a) Cisco ASR 1000 Series [14]; (b) Cisco 3945 ISR [15].

III. Problem Formulation

A. System Background

The bandwidth reservation service in HPNs follows two fundamental processes: resource scheduling and path routing based on resource availability and existing reservations, i.e., multi-constrained path computation. In real-life bandwidth reservation services such as OSCARS [1] of DOE's ESnet as illustrated in Fig. 2 [34], the resource manager typically maintains *a priori* knowledge of exact flow cost and an accurate snapshot of available network resources, which are used to determine the best reservation option for a given user request.

B. Cost Models

We consider an HPN $G(V, L)$ that consists of a set V of routers connected through a set L of full duplex wired links of capacities $C_L = \{C_l | l \in L\}$. Each router v is equipped with N_v line cards $c_{v,i}$, $i = 1, 2, ..., N_v$, each of which contains multiple ports [28]. Each line card $c_{v,i}$ includes a transmitter $c^T_{v,i}$ and a receiver $c^R_{v,i}$. The set V of routers and the set LC of line cards on all the routers make up a set D of network devices, i.e. $D = V \cup LC$. Each pair of adjacent routers are connected by one port-to-port link, which is associated with a link error rate γ_l ($0 \leq \gamma_l \leq 1$). We

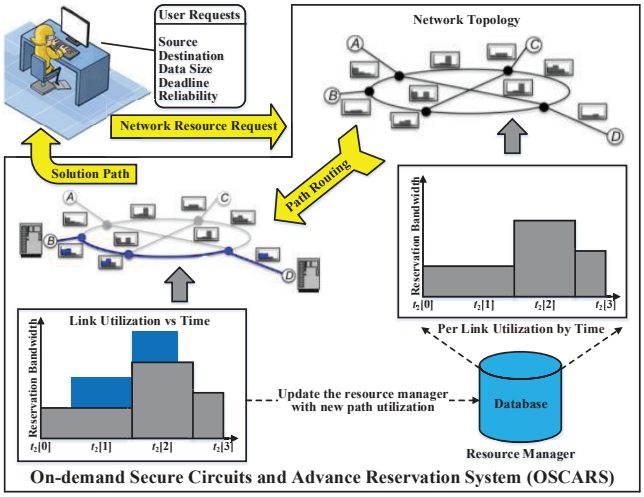


Fig. 2. On-demand Secure Circuits and Advance Reservation System.

introduce a path transmission failure rate (PTFR) γ_p to define the reliability of data transfer along a path, which is measured as the accumulated error rate over all component links of a path p, i.e. $\gamma_p = 1 - \prod_{l \in p}(1-\gamma_l)$. A user data transfer request $R(v_s, v_d, \delta, t^A, t^D, \gamma)$ specifies the source v_s, the destination v_d, the data size δ, the time point t^A ($t^A \geq 0$) when the data becomes available for transfer, the deadline t^D by which the transfer must be completed, and the maximum allowed PTFR γ of the data transfer.

1) Bandwidth Reservation Model

We use a 3-tuple of time-bandwidth (TB) $(t_l[i], t_l[i+1], b_l[i])$ to represent the available (or residual) bandwidth $b_l[i]$ of link l at time interval $[t_l[i], t_l[i+1]), i = 0, 1, \cdots, T_l - 1$, where T_l is the total number of time slots of link l. For each directed link $l \in L$, its TB list corresponds to a step function with respect to time t to describe the available bandwidth of link l. We combine the TB lists of all links to build an Aggregated TB (ATB) table, where we store the available bandwidths of all links in each intersected time slot. As shown in Fig. 3, we first create a set of new time slots by combining the time slots of all links $l \in L$, and then map the available bandwidths of each link to the ATB table in each new time slot. We denote the ATB table as $(t[0], t[1], b_0[0], b_1[0], \cdots, b_{m-1}[0]), \cdots, (t[T_A - 1], t[T_A], b_0[T_A - 1], b_1[T_A - 1], \cdots, b_{m-1}[T_A - 1])$, where T_A is the total number of new time slots after the aggregation of TB lists of m links. The time slot i corresponds to the time interval $[t[i], t[i+1])$, and $t[T_A] = +\infty$ [24]. The scheduler maintains the ATB table $B_L^A(t)$ for all directed links L in all future time slots.

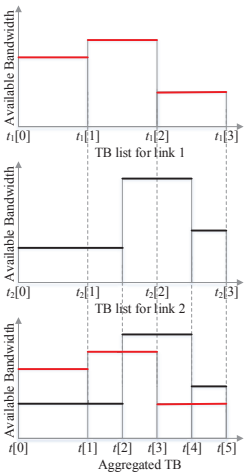


Fig. 3. An example ATB table.

Once the scheduler admits a new user request over a computed network path for a certain time duration, or a network device is shut down or booted up, the ATB is dynamically updated: $B_l^A(t) = C_l(t) - B_l^R(t) \geq 0$, where $C_l(t)$ is the actual capacity of link l at time point t, and $B_l^R(t)$ is the reserved bandwidth on link l at time point t. We use $B_p(t)$ to denote the bottleneck bandwidth of path $p_{s,d}$ from source v_s to destination v_d at time point t, i.e. $B_p(t) = \min_{l \in p} B_l^A(t)$ [28].

2) Power Models

According to the architecture of commercial routers, we extend the power model in [11] to a more general one, and use it to calculate the power consumption of router v with a set of traffic loads $R_v = \{r_v, r_{v,i}^T, r_{v,i}^R | i = 1, \cdots, N_v\}$, where r_v is the sum of incoming, outgoing and forwarding data rates on router v, and r_i^T and r_i^R are transmitted and received data rates on line card $c_{v,i}$, respectively, i.e. $P_v = P_v^S + P_v^D(R_v)$, where $P_v^S = \sum_{i=0}^{N_v} P_{v,i}^S$, and $P_v^D(R_v) = P_{v,0}^D(r_v) + \sum_{i=1}^{N_v} P_{v,i}^T(r_{v,i}^T) + \sum_{i=1}^{N_v} P_{v,i}^R(r_{v,i}^R)$. Here, P_v^S is the static power consumption (SPC) of router v, which consists of the SPC $P_{v,0}^S$ by the chassis of router v and the SPC of all the line cards on router v; $P_{v,i}^S$ ($i \geq 1$) is the SPC of the i-th line card on router v in a base configuration. $P_v^D(R_v)$ is the DPC of router v, which is a function of traffic loads R_v. $P_{v,0}^D(r_v)$ is the DPC function for the chassis of router v with respect to the total traffic load r_v. $P_{v,i}^T(r_{v,i}^T)$ and $P_{v,i}^R(r_{v,i}^R)$ are the DPC functions of transmitting rate $r_{v,i}^T$ and receiving rate $r_{v,i}^R$ on line card $c_{v,i}$, respectively. Note that wide-area connections may require relays, which also consume energy. However, such energy consumption could be combined with line cards at two ends, and hence is not considered here. Accordingly, the incremental energy $E_p(t_1, t_2)$ consumed by a user request over path p during a time window from time t_1 to time t_2 consists of a static part $E_p^S(t_1, t_2)$ and a dynamic part $E_p^D(t_1, t_2)$:

$$E_p(t_1, t_2) = E_p^S(t_1, t_2) + E_p^D(t_1, t_2),$$

where

$$E_p^D(t_1, t_2) = \int_{t_2}^{t_1} \Bigg\{ \sum_{v \in p} [P_{v,0}^D(r_v(t) + r_p(t_1, t_2)) - P_{v,0}^D(r_v(t))] + \sum_{c_{v,i}^T \in p} [P_{v,i}^T(r_{v,i}^T(t) + r_p(t_1, t_2)) - P_{v,i}^T(r_{v,i}^T(t))] + \sum_{c_{v,i}^R \in p} [P_{v,i}^R(r_{v,i}^R(t) + r_p(t_1, t_2)) - P_{v,i}^R(r_{v,i}^R(t))] \Bigg\} dt. \quad (1)$$

Eq. 1 calculates the incremental DEC of all routers and line cards on a path. $E_p^S(t_1, t_2)$ is specific to the power model in use and will be defined later in this subsection for different power models.

We present the two main power models as follows.

i) Power-Down (PD) Model

The scheduler uses a **booting-up time (BUT) list** T_D^U (including entries $T_{v,0}^U$ and $T_{v,i}^U$) and a **shutting-down time (SDT) list** T_D^D (including entries $T_{v,0}^D$ and $T_{v,i}^D$) to record the amount of time required for activating and deactivating each network device (including the chassis of each router v

and each line card $c_{v,i}$), respectively. Based on T^U_D, T^D_D and the current bandwidth reservation status, we are able to make our bandwidth scheduler energy-aware by shutting down idle network devices if the resulted energy saving is greater than the energy consumed for bringing them back up. Also, we use an **SPC table** $P^S_D(t)$ to keep track of the time-varying SPC of each network device based on the ATB, BUT, and SDT tables. At time point t, for the chassis of router v (or line card $c_{v,i}$) in one of the four different states, i.e. two stable states (powered-off and active) and two transitional states (booting-up and shutting-down), we set its corresponding entry $P^S_{v,0}(t)$ (or $P^S_{v,i}(t)$) in the SPC table to be 0, $P^{AS}_{v,0}$ (or $P^{AS}_{v,i}$), $P^{US}_{v,0}$ (or $P^{US}_{v,i}$), and $P^{DS}_{v,0}$ (or $P^{DS}_{v,i}$), respectively. In this model, we calculate the incremental SEC of all affected routers and line cards for transferring a new bulk of data from t_1 to t_2 along path p as follows:

$$E^S_p(t_1,t_2) = \sum_{c_{v,i}\in p} E^S_{v,i}(t_1,t_2) + \sum_{v\in p} E^S_{v,0}(t_1 - \max_{c_{v,i}\in p} T^U_{v,i}, t_2 + \max_{c_{v,i}\in p} T^D_{v,i}), \quad (2)$$

where

$$\begin{aligned} E^S_{v,i}(t_1,t_2) &= P^{AS}_{v,i}\cdot(t_2-t_1) \\ &+ \min\{P^{AS}_{v,i}\cdot t^F_{v,i}(t_1), P^{US}_{v,i}\cdot T^U_{v,i} + P^{DS}_{v,i}\cdot T^D_{v,i}\} \\ &+ \min\{P^{AS}_{v,i}\cdot t^B_{v,i}(t_2), P^{US}_{v,i}\cdot T^U_{v,i} + P^{DS}_{v,i}\cdot T^D_{v,i}\} \\ &- \int_{t_1-t^F_{v,i}(t_1)}^{t_2+t^B_{v,i}(t_2)} P^S_{v,i}(t)dt. \end{aligned} \quad (3)$$

Here, $E^S_{v,0}(t_1,t_2)$ and $E^S_{v,i}(t_1,t_2)$ $(i\geq 1)$ are the incremental static energy consumed by the chassis of router v and the i-th line card on router v, respectively, to transfer data from t_1 to t_2 for a given user request; $t^F_D(t)$ and $t^B_D(t)$ are the shortest period from the previous time point when the network device is active to time point t, and the shortest period from time point t to the subsequent time point when the network device is active, respectively. In Eq. 2, to guarantee that the transmitting and receiving line card(s) are powered on (i.e. active/booting up/shutting down) from t_1 to t_2, the chassis of router v should be active from $t_1 - \max_{c_{v,i}\in p} T^U_{v,i}$ to $t_2 + \max_{c_{v,i}\in p} T^D_{v,i}$, and would consume $E^S_{v,0}(t_1 - \max_{c_{v,i}\in p} T^U_{v,i}, t_2 + \max_{c_{v,i}\in p} T^D_{v,i})$ more static energy. To further clarify the calculation of the incremental SEC of a network device, we provide an illustrating example in Fig. 4 to explain Eq. 3 as follows. At the current time $t=0$, the SPC $P^S_{v,i}(t)$ of a line card $c_{v,i}$ in the SPC table is shown in the upper part of Fig. 4 based on existing bandwidth reservations. If we transfer data for a new user request from t_1 to t_2 via $c_{v,i}$, $c_{v,i}$ should be active during the period of $[t_1,t_2]$, and would consume static energy corresponding to the first term in Eq. 3 and the green area with vertical lines in Fig. 4. To ensure that $c_{v,i}$ is ready to use at t_1, we check its preceding period of $[t_1 - t^F_{v,i}(t_1), t_1]$ to decide whether $c_{v,i}$ should remain continuously active during the period or be first shut down and then booted up for possible energy saving. Based on the decision, the SEC of $c_{v,i}$ during the preceding period corresponds to the second term in Eq. 3 and the blue area with horizontal lines in Fig. 4. Similarly, the SEC of $c_{v,i}$ based on its status selection during the succeeding period of $[t_2, t_2 + t^B_{v,i}(t_2)]$ corresponds to the third term in Eq. 3 and zero area in Fig. 4, where $t^B_{v,i}(t_2) = 0$. In addition, if we do not transfer any new data, the SEC of $c_{v,i}$ during the entire period of $[t_1 - t^F_{v,i}(t_1), t_2 + t^B_{v,i}(t_2)]$ corresponds to the fourth term in Eq. 3 and the orange area with slanted lines in Fig. 4. As shown in Eq. 3, the incremental SEC of $c_{v,i}$ is the difference between the SEC of $c_{v,i}$ during the period of $[t_1 - t^F_{v,i}(t_1), t_2 + t^B_{v,i}(t_2)]$ with and without the transfer of new data.

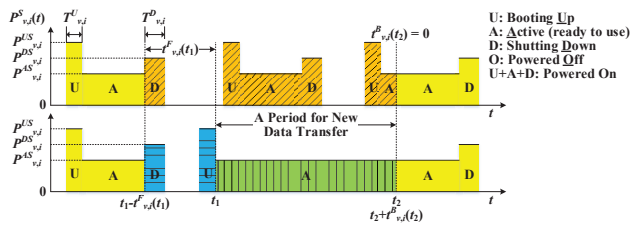
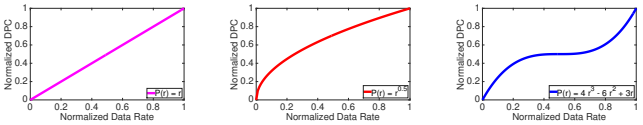


Fig. 4. An example for the power-down model.



(a) (b) (c)

Fig. 5. Different dynamic power modeling functions: (a) a linear function; (b) a concave function; (c) a polynomial function.

The scheduler maintains a **powered-on time (POT) list** T^{on}_D (including entries T^{on}_v and $T^{on}_{v,i}$) to record the amount of time, during which a network device (such as router v and line card $c_{v,i}$) is continuously powered on up to the current time point. During booting-up, the router cannot be used and the lines cards cannot be activated until the router is active, i.e. $T^{on}_{v,i} = 0$ if $T^{on}_v \leq T^U_{v,0}$.

Based on the BUT, SDT, and POT lists, the bandwidth scheduler calculates an available state table $A_D(t)$ as follows: if router v or line card $c_{v,i}$ is active at time point t, we set $A_v(t)$ or $A_{v,i}(t)$ to be 1; otherwise, we set it to be 0. Obviously, $A_{v,i}(t) = 0$ if $A_v(t) = 0$. The actual capacity of network link l at time point t depends on the availability status of the line cards $c_{v,i}, c_{u,j} \in l$ on both ends of the link, i.e. $C_l(t) = C_l \cdot A_{v,i}(t) \cdot A_{u,j}(t)$ [28].

Each network device has constant SPC $P^S_{v,i} > 0$ while powered on. Here, we model DPC $P^D_v(r_v)$, $P^T_{v,i}(r^T_{v,i})$ and $P^R_{v,i}(r^R_{v,i})$ by non-decreasing concave functions $P(r) \geq 0$ of data rate r with $P(0) = 0$, as shown in Figs. 5(a) and 5(b).

ii) Speed-Scaling (SS) Model

In this model, each router $v \in V$ and its line cards remain powered on, so there is no incremental SEC for a data transfer request, i.e. $E^S_p(t_1,t_2) = 0$. We use a generalized polynomial function $P(r)$ of data rate r to model the DPC of different network devices, i.e. power modeling functions $P^D_v(r_v)$, $P^T_{v,i}(r^T_{v,i})$ and $P^R_{v,i}(r^R_{v,i})$ of data rates r_v, $r^T_{v,i}$, and $r^R_{v,i}$ for the chassis, line card transmitters and receivers, respectively. The power modeling function $P(r)$ is non-decreasing in R_+ (i.e. $dP(r)/dr \geq 0$ for $r \in R_+$), in the form of $P(r) = a_k r^k + a_{k-1} r^{k-1} + \cdots + a_1 r, a_i \in R, k \in Z_{++}$, as shown in Fig. 5(c).

TABLE I
NOTATIONS USED IN THE COST MODELS.

Notations	Definitions
$G(V, L)$	A directed network graph of a set V of routers and a set L of directed links among them
C_L	A set of the capacities of directed links L
N_v	The number of line cards on router v
LC	A set of line cards
$c_{v,i}$, $c^T_{v,i}$, and $c^R_{v,i}$	The i-th line card on router v and its transmitter and receiver, respectively
D	A set of network devices (routers and line cards)
γ_L	A set of link error rates of directed links L
$L^{out}(v)$ and $L^{in}(v)$	A set of outgoing and incoming directed links from router v, respectively
$R(v_s, v_d, \delta, t^A, t^D, \gamma)$	A user request for transferring data of size δ from source v_s to destination v_d with PTFR $\leq \gamma$ after data available time t^A before deadline t^D
T^U_D and T^D_D	The booting-up and shutting-down time lists of all devices, respectively
P^{AS}_D, P^{US}_D, and P^{DS}_D	The active, booting-up and shutting-down SPC list of all devices, respectively
$P^S_D(t)$	The SPC list of all devices at time t without the current data transfer request
$t^F_v(t)$ (or $t^F_{v,i}(t)$)	The shortest period from the previous time point when router v (or line card $c_{v,i}$) is powered on to the time point t
$t^B_v(t)$ (or $t^B_{v,i}(t)$)	The shortest period from the time point t to the subsequent time point when router v (or line card $c_{v,i}$) is powered on
$B^A_l(t)$	The available bandwidth of directed link l at time t
$B_p(t)$	The bottleneck bandwidth of path p at time t
$P_{s,d}$	A set of paths from source v_s to destination v_d
$r_p(t_1, t_2)$	The data rate on path p from time t_1 to time t_2
$r_v(t)$	The total data rate of incoming, outgoing and forwarding flows on router v at time t
$r^T_{v,i}(t)$ and $r^R_{v,i}(t)$	The total data rate of outgoing and incoming flows on line card $c_{v,i}$ at time t, respectively
$P^S_{v,0}$	The SPC of the chassis of router v
$P^S_{v,i} (i \geq 1)$	The SPC of line card $c_{v,i}$
$P^D_{v,0}(r)$	The DPC of the chassis of router v with traffic loads at data rate r
$P^T_{v,i}(r)$ and $P^R_{v,i}(r)$ $(i \geq 1)$	The DPC of line card $c_{v,i}$ for transmitting and receiving data at rate r, respectively
$E_p(t_1, t_2)$	The incremental energy consumption over path p from time t_1 to time t_2
$E^S_p(t_1, t_2)$ and $E^D_p(t_1, t_2)$	The incremental SEC and DEC over path p from time t_1 to time t_2, respectively

We tabulate the main notations used in the cost models in Table I for convenience.

C. Problem Definition

Based on the above cost models, we formulate an instant bandwidth scheduling problem using PD and SS models, respectively, in high-performance networks. In view of different transport constraints (i.e. a fixed path or varying paths) and application requirements (i.e. a fixed data rate/bandwidth or varying data rates/bandwidths), Lin *et al.* categorized instant bandwidth scheduling into four types [24]. One of these types is to establish a channel along a fixed network path from a source node to a destination node with a fixed bandwidth, referred to as FPFB, during the whole period of a given data transfer. Since FPFB is one of the most commonly used service models in real-life HPNs, we formulate our problem based on FPFB as follows:

Definition 1. ***FPFB-MEC****: Given an HPN $G(V, L)$ where each link $l \in L$ is associated with a link capacity C_l, traffic load $r_l(t)$, and a link error rate γ_l, as well as a user request $R(v_s, v_d, \delta, t^A, t^D, \gamma)$, based on the following two power models:*

- ***Power-Down (PD)****: booting-up and shutting-down time lists T^U_D and T^D_D, an SPC table $P^S_D(t)$, a powered-on time list T^{on}_D, as well as SPC $P^{AS}_{v,i}$, $P^{US}_{v,i}$, and $P^{DS}_{v,i}$ and dynamic power models $P^D_v(r_v)$, $P^T_{v,i}(r^T_{v,i})$ and $P^R_{v,i}(r^R_{v,i})$ in the form of concave functions,*
- ***Speed-Scaling (SS)****: polynomial power modeling functions $P^D_v(r_v)$, $P^T_{v,i}(r^T_{v,i})$ and $P^R_{v,i}(r^R_{v,i})$ of data rates r_v, $r^T_{v,i}$, $r^R_{v,i}$ for the chassis, line card transmitters and receivers, for each router $v \in V$ with the line cards always powered on,*

we wish to find a triplet (p, t_1, t_2) of a fixed path p with a fixed data rate r_p, start time t_1, and end time t_2 to meet the user request R with minimum energy consumption:

$$\min_{p \in P_{s,d}; t_1; t_2} E_p(t_1, t_2),$$

subject to

$$t^A \leq t_1 < t_2 \leq t^D,$$
$$(t_2 - t_1) \cdot r_p(t_1, t_2) = \delta,$$
$$r_p(t_1, t_2) \leq \min_{t_1 \leq t \leq t_2} B_p(t),$$
$$\prod_{l \in p} (1 - \gamma_l) \geq 1 - \gamma.$$

IV. COMPLEXITY ANALYSIS

A. Complexity Analysis for FPFB-MEC-PD

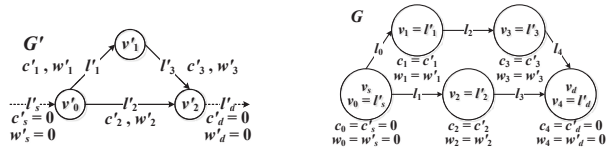
We prove that FPFB-MEC-PD is NP-complete by reducing from the restricted shortest path (RSP) problem, whose NP-completeness is shown in [18].

Theorem 1. *FPFB-MEC-PD is NP-complete.*

Proof. Obviously, FPFB-MEC-PD $\in$ NP. Given a solution to FPFB-MEC-PD, one can verify the validity of the solution in polynomial time by checking whether or not the data transfer meet the requirements and the energy consumption bound for the decision version of FPFB-MEC-PD.

We first consider a special case of the decision version of FPFB-MEC-PD with a bound E on the energy consumption as follows: In the user request, $t^D = t^A + \delta/C$; in the HPN, all the links are idle with identical link capacity C after t^A and there is no other power consumption except the SPC $\{P^S_{v,0}\}$ of the router chassis. Obviously, $t_1 = t^A$ and $t_2 = t^A + \delta/C$. The special case is equivalent to the following problem (P1): Given a directed graph $G(V, L)$ with v_s and v_d, a non-negative weight $w_l = -log(1 - \gamma_l)$ for each directed edge, and a non-negative cost $c_v = P^S_{v,0} \cdot \delta/C$ for each node, the goal is to find the path p from v_s to v_d under the total weight constraint $\sum_{l \in p} w_l \leq W = -log(1-\gamma)$ with the total cost no more than E. Furthermore, for any $i, j \in L$, let $w_i = w_j = w_v$ if edges i and j end at the same node v. Accordingly, a problem P2 as a special case of P1 can be stated as follows: Given $G(V, L)$ with v_s and v_d, non-negative weight w_v and cost c_v for each node, the goal is to find a path with the total cost $\sum_{v \in p} c_v \leq E$ under the total weight constraint $\sum_{v \in p, v \neq s} w_v \leq W$.

We now reduce to P2 from RSP, which is defined as follows: Given a directed graph $G'(V', L')$ with source v'_s, destination v'_d, cost $c'_{l'} \geq 0$ and weight $w'_{l'} \geq 0$ for each edge, the goal is to find the path with the total cost no more than E from v'_s to v'_d subject to a limit W on the total weight. For any instance in RSP, we construct an instance of P2 by adding two new virtual directed edges $\{l'_s, l'_d\}$ into $G'(V', L')$, where l'_s and l'_d are an incoming link to v_s and an outgoing link from



(a) (b)

Fig. 6. A small network example: (a) a network graph G' in RSP; (b) the newly constructed network graph G in P2.

v_d, respectively, and then constructing a new directed graph $G(V, L)$ with $V = L' \cup \{l'_s, l'_d\}$. If the end node of l'_i and the start node of l'_j are the same in G', there is a directed edge from node l'_i to node l'_j in G. For illustration, we provide a small network example in Fig. 6(a), where $V' = \{v'_0, v'_1, v'_2\}$, $L' = \{l'_1, l'_2, l'_3\}$, $v'_s = v'_0$ and $v'_d = v'_2$. In the newly constructed graph, $V = \{l'_1, l'_2, l'_3, l'_s, l'_d\}$, $L = \{l_0, l_1, l_2, l_3, l_4\}$, and node costs and weights are shown in Fig. 6(b). Obviously, this construction can be done in polynomial time, and a solution to the RSP or P2 instance can solve the other. Hence, RSP is reduced to P2, and thus to P1 and FPFB-MEC-PD. Since RSP is NP-complete, so is FPFB-MEC-PD. Proof ends. □

B. Complexity Analysis for FPFB-MEC-SS

We also prove that FPFB-MEC-SS is NP-complete by reducing from RSP to a special case of FPFB-MEC-SS.

Theorem 2. *FPFB-MEC-SS is NP-complete.*

Proof. Obviously, FPFB-MEC-SS $\in NP$. We consider a special case of the decision version of FPFB-MEC-SS as follows: In the user request, $t^D = t^A + \delta/C$; in the HPN, all the links are active and idle with identical link capacity C after t^A and each router only incurs linear DPC $P_v^D(r_v) = f_v \cdot r_v$. Obviously, $t_1 = t^A$ and $t_2 = t^A + \delta/C$. The special case is equivalent to the following problem (P3): Given $G(V, L)$ with v_s and v_d, a non-negative weight $w_l = -log(1 - \gamma_l)$ for each directed edge, and a non-negative cost $c_v = f_v \cdot \delta$ for each node, the goal is to find path p with the total cost no more than E under the total weight constraint $\sum_{l \in p} w_l \leq W = -log(1 - \gamma)$. Since P3 is equivalent to P1, which has been proved to be NP-complete in the proof of Theorem 1, FPFB-MEC-SS is also NP-complete. Proof ends. □

V. Algorithm Design for FPFB-MEC-PD

A. An FPTAS for FPFB-MEC-PD

1) Algorithm Framework

We propose an FPTAS, named SAVEE-PD-App, for the FPFB-MEC-PD problem. The pseudocode of SAVEE-PD-App is provided in Alg. 1, which follows the framework of the algorithm design of the optimal solution SAVEE-PD-Opt for the special case of FPFB-MEC-PD where the PTFR constraint is sufficiently large (i.e. no reliability constraint) in [28].

Given a user request, the algorithm first updates the ATB table according to the BUT and POT lists (Line 1). T_{SPC} contains the start and end time points of all the time slots in the SPC table (Line 2); and T_U (or T_D) contains the booting-up (or shutting-down) time of all the line cards when the router is active or powered off (Line 3). Since the data transfer must start and finish within the time period from t^A to t^D, the algorithm varies the transfer start time slot $x+1$ from the first time slot to the y-th time slot for a given data transfer end time slot y, and finds the path ρ with the minimum energy consumption such that the data of size δ can be transferred during the time window of slots $[x+1, y]$. Here, a time window is defined as a set of contiguous time slots. The algorithm repeatedly increases y by 1, and computes the approximately optimal transfer start time t_1 and end time t_2 by considering all possible x and y values (Lines 4-5).

We define the following notations in SAVEE-PD-App: i) b_l: the maximum available bandwidth of link l over the time window of slots $[x+1, y]$ (Line 7), ii) B_0: the minimum bandwidth to transfer the data of size δ during the time window of slots $[x+1, y]$ (Line 8), iii) B_1: the maximum bandwidth to transfer the data of size δ from the beginning of the start time slot $x+1$ to a time point in the end time slot y (Line 8), iv) B_2: the maximum bandwidth to transfer the data of size δ from a time point in the start time slot $x+1$ to the end of the end time slot y (Line 17). In each time window, SAVEE-PD-App selects all the available link bandwidths within the upper and lower boundaries as mentioned above (Lines 9, 18). For a given path and a given bandwidth, an optimal bandwidth reservation either starts at time $t_s \in T_S$ (T_S is the set of all possible optimal data transfer start time points as defined in Line 10) or ends at time $t_e \in T_E$ (T_E is the set of all possible optimal data transfer end time points as defined in Line 19), depending on the SEC of data transfer over different time slots in the SPC table. Hence, SAVEE-PD-App calculates the exact time window $[\tau_1, \tau_2]$ (Lines 12, 21). Then, the minimum energy consumption ε in this time window and the corresponding path are calculated using an existing FPTAS in [17] or [12] based on a new graph constructed from the original one (Lines 13-14, 22). SAVEE-PD-App examines all possible optimal transfer time windows and bandwidths, and guarantees that the returned energy consumption is within an upper bound (Lines 9-16, 18-22). Finally, SAVEE-PD-App updates the ATB table to reserve the bandwidth, and the SPC table to boot up necessary devices and shut down unused devices (Line 23).

2) Network Graph Reconstruction for Shared SEC

In the original network graph, the number of all possible paths is exponential, which prohibits an exhaustive search in large-scale networks. Since routers and line cards are the main energy consumers, we associate each router equipped with ingress and egress line cards with a cost of energy consumption (not a link cost). Therefore, the shortest path algorithm does not work on the original network graph.

To address this issue, we construct a new directed graph $G'(V', L')$ with weight of $w_{L'}$ and cost of $c_{L'}$ from $G(V, L)$ (Line 13), where the links with bandwidth less than the requirement β, the incoming links to v_s and outgoing links from v_d have been removed from L as they do not affect the problem. Here, the set $V' = L \cup \{l_s, l_d\}$, where l_s is a virtual link to source v_s and l_d is a virtual link from destination v_d in G. The set L' contains pairs of links in G that are connected without forming a loop, and each edge l' can be represented by a triplet of a router v and its ingress and egress line cards

Algorithm 1: SAVEE-PD-App

Input: G, C_L, γ_L, R, ATB, SPC, BUT, POT and $\{P^S_{v,i}\}$
Output: Energy $E_{\min}$, path p, start time t_1, end time t_2

1: Updates the ATB table according to the BUT and POT lists;
2: $T_{SPC} = \{$the start and end time points of all the time slots in the P^S_D table$\}$;
3: $T_U = \{0, T^U_{v,i}, T^U_{v,i} + T^U_{v,0} | v \in V, c_{v,i} \in LC\}$;
$T_D = \{0, T^D_{v,i}, T^D_{v,i} + T^D_{v,0} | v \in V, c_{v,i} \in LC\}$; $E_{\min} = \infty$;
4: **for** $y = 1$ to T_A **do**
5: **for** $x = 0$ to $y - 1$ **do**
6: **for all** $l \in L$ **do**
7: $b_l = \min_{x \le i \le y-1} b_l[i]$;
8: $B_0 = \frac{\delta}{t[y]-t[x]}$; $B_1 = \frac{\delta}{t[y-1]-t[x]}$;
9: **for all** $\beta \in \{b_l | B_0 \le b_l < B_1, l \in L\}$ **do**
10: $T_S = \{t'_1 + t'_2 | t[x] \le t'_1 + t'_2 \le t[y] - \frac{\delta}{\beta}, t'_1 \in T_{SPC}, t'_2 \in T_U\}$;
11: **for all** $t_s \in T_S$ **do**
12: $\tau_1 = t_s$; $\tau_2 = t_s + \frac{\delta}{\beta}$;
13: Construct a directed graph $G'(V', L')$ from G with weights $w_{L'}$ and costs $c_{L'} = \{E_{l'}(\tau_1, \tau_2), l' \in L'\}$;
14: $(\varepsilon, \rho) =$ the minimum energy cost and the corresponding path to transfer the data of size δ from v_s to v_d during time window $[\tau_1, \tau_2]$ on G';
15: **if** $\varepsilon < E_{\min}$ **then**
16: $E_{\min} = \varepsilon$; $p = \rho$; $t_1 = \tau_1$; $t_2 = \tau_2$;
17: $B_2 = \frac{\delta}{t[y]-t[x+1]}$;
18: **for all** $\beta \in \{b_l | B_0 \le b_l < B_2, l \in L\}$ **do**
19: $T_E = \{t'_3 - t'_4 | t[x] + \frac{\delta}{\beta} \le t'_3 - t'_4 \le t[y], t'_3 \in T_{SPC}, t'_4 \in T_D\}$;
20: **for all** $t_e \in T_E$ **do**
21: $\tau_1 = t_e - \frac{\delta}{\beta}$; $\tau_2 = t_e$;
22: Compute the minimum energy consumption and the corresponding path (ε, ρ) in the same way as above, and update $E_{\min}$, p, t_1 and t_2 if needed;
23: Update the ATB table and the SPC table to indicate when the routers and line cards are booted up or shut down.
24: **return** $(E_{\min}, p, t_1, t_2)$.

$c_{v,i}, c_{v,e}$ that connect the pair of directed links in G. The cost $c_{l'}$ of edge $l' \in L'$ is the incremental energy consumption of the triplet $(v, c_{v,i}, c_{v,e})$ for transferring data of size δ from τ_1 to τ_2, which is a constant at a given data rate within a given time window. The weight $w_{l'}$ of edge $l' \in L'$ is the same as the weight $w_l = -log(1 - \gamma_l)$, where l is the starting vertex of l'. For illustration, we provide a small example in Fig. 7(a), where $V = \{v_0, v_1, v_2, v_3\}$, L is updated from $\{l_0, l_1, \ldots, l_9\}$ to $\{l_0, l_1, \ldots, l_5\}$ as $v_s = v_0$ and $v_d = v_3$, and the line card configurations are as follows: l_0 is from $c_{0,1}$ in v_0 to $c_{1,1}$ in v_1; l_1 is from $c_{0,2}$ in v_0 to $c_{2,1}$ in v_2; l_2 is from $c_{1,2}$ in v_1 to $c_{2,2}$ in v_2; l_3 is from $c_{2,2}$ in v_2 to $c_{1,2}$ in v_1; l_4 is from $c_{1,1}$ in v_1 to $c_{3,1}$ in v_3; l_5 is from $c_{2,1}$ in v_2 to $c_{3,2}$ in v_3. In the new graph constructed from the original one, $V' = \{l_0, l_1, \ldots, l_5, l_s, l_d\}$ and $L' = \{l'_0, l'_1, \ldots, l'_9\}$ as shown in Fig. 7(b).

B. Performance Bound and Time Complexity Analysis of SAVEE-PD-App

Based on the optimality proof and the time complexity analysis for SAVEE-PD-Opt in [28], we derive the approximate ratio of SAVEE-PD-App.

Theorem 3. *SAVEE-PD-App finds a feasible triplet (p, t_1, t_2) of energy consumption within the least energy consumption multiplied by $(1 + \epsilon)$ in time $O(T^3_A |L|^2 |D|^3 (\log\log\log |D| + \epsilon^{-1}))$ if the FPTAS in [17] is used in Lines 14 and 22 in Alg. 1.*

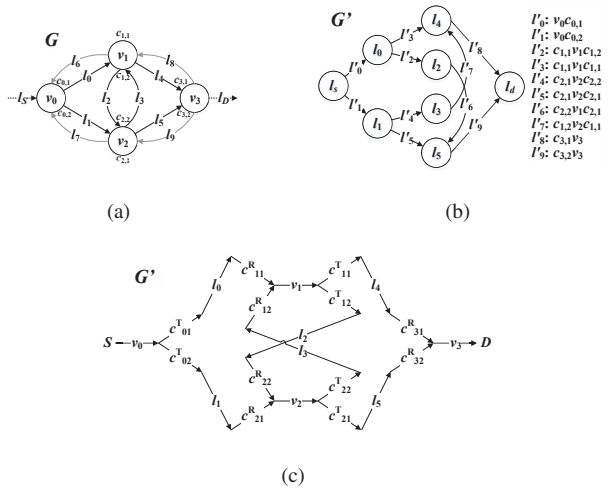


Fig. 7. A small network example: (a) the original network graph; (b) the newly constructed graph for the power-down model; (c) the newly constructed graph for the speed-scaling model.

Proof. The FPTAS of RSP in [17] can find a feasible path whose cost is within the least cost multiplied by $(1+\epsilon)$ in time $O(mn(\log\log\log n + \epsilon^{-1}))$, where n and m are the number of nodes and the number of edges, respectively. The approximate ratio of SAVEE-PD-App is the worst one in all the cases. Proof ends. □

Theorem 4. *SAVEE-PD-App finds a triplet (p, t_1, t_2) with no more than the least energy consumption and PTFR within $(1 + \epsilon)$ of its requirement in time $O(T^3_A |L||D|^3 (|L| + |D| log |D|) \epsilon^{-1})$ if the FPTAS in [12] is used in Lines 17 and 27 in Alg. 1.*

Proof. The FPTAS of RSP in [12] can find a path with no more than the least cost and a weight within $(1 + \epsilon)$ of its requirement in time $O((m + n \log n) H \epsilon^{-1})$, where n, m and H are the number of nodes, the number of edges, and the number of hops of the longest unweighted path, respectively. Proof ends. □

VI. Algorithm Design for FPFB-MEC-SS

In this section, we design a pseudo-polynomial time approximation algorithm and a heuristic algorithm for FPFB-MEC-SS.

Since the transmitter and receiver of a line card are separate components with respective dynamic power functions, we consider physical links, router chassis, as well as line card transmitters and receivers as directed edges in G', i.e. $L' = L \cup V \cup \{c^T_{v,i}, c^R_{v,i} | c_{v,i} \in LC\}$. The cost $c_{l'}$ of edge $l' \in V \cup \{c^T_{v,i}, c^R_{v,i} | c_{v,i} \in LC\}$ is the incremental energy consumption of network device v, $c^T_{v,i}$, $c^R_{v,i}$ for transferring data of size δ from time τ_1 to time τ_2, which is a constant at a given data rate within a given time window. Ignoring the energy consumption of fibers, we have $c_{l'} = 0$ for $l' \in L'$. The weight $w_{l'}$ of edge $l' \in L'$ is the same as the weight $w_l = -log(1 - \gamma_l)$, and $w_{l'} = 0$ for $l' \in V \cup \{c^T_{v,i}, c^R_{v,i} | c_{v,i} \in LC\}$. For illustration, we provide a small network example in Fig. 7(a) as mentioned in Subsection V-A. In the new graph constructed from the original one, $L' = \{v_0, v_1, v_2, v_3\} \cup$

$\{c^T_{0,1}, c^T_{0,2}, c^T_{1,1}, c^T_{1,2}, c^T_{2,1}, c^T_{2,1}, c^R_{1,1}, c^R_{1,2}, c^R_{2,1}, c^R_{2,1}, c^R_{3,1}, c^R_{3,2}\} \cup \{l_0, l_1, l_2, l_3, l_4, l_5\}$ as shown in Fig. 7(c).

Furthermore, we design algorithms for FPFB-MEC-SS based on the following lemma.

Lemma 1. *If FPFB-MEC-SS has feasible solutions, there exists an optimal solution where either the start time is the beginning of a time slot in the ATB table or the end time is the end of a time slot in the ATB table.*

Proof. The energy consumption of data transfer starting at the J-th time slot and ending at the K-th time slot on link l' is

$$\begin{aligned} &E_{l'}(t_1, t_1 + \delta r^{-1}) = \\ &\sum\nolimits_{j=J+1}^{K-1} (t[j] - t[j-1])(P_{l'}(r + r_{l'}[j]) - P_{l'}(r_{l'}[j])) \\ &+ (t[J] - t_1)(p_{l'}(r + r_{l'}[J]) - p_{l'}(r_{l'}[J]))+ \\ &(t_1 + \delta r^{-1} - t[K-1])(p_{l'}(r + r_{l'}[K]) - p_{l'}(r_{l'}[K])). \end{aligned} \tag{4}$$

From Eq. 4, $\sum_{l' \in p} E_{l'}(t_1, t_1 + \delta r^{-1})$ is a linear function of t_1. Hence, given a data rate r, the energy consumption is minimized within the time window of slots $[J, K]$ when $t_1 = t[J-1]$ or $t_1 = t[K] - \delta r^{-1}$. Proof ends. □

A. An Approximation Algorithm for FPFB-MEC-SS

1) Key Idea for Dynamic Energy Saving

We exploit the feature of a non-monotonously increasing dynamic energy consumption function derived from the polynomial power function to design a pseudo-polynomial ϵ-approximation algorithm, named SAVEE-SS-App, for FPFB-MEC-SS in Alg. 2. Since the DPC function is a non-decreasing polynomial function, the incremental DEC function (IDEC) derived from it might decrease with respect to data rate, but not in a drastic manner. In other words, the derivative of IDEC has an upper bound, which is expressed in a simple inequality in Lemma 2 for a special case. As a result, the key idea of our approximation algorithm is to select a series of data rate samples at a uniform interval ξ, and then compute the path corresponding to each data rate sample. Note that the upper bound of the sampling interval would limit the difference between the optimal IDEC and the best IDEC calculated from all the samples.

2) Performance Bound and Time Complexity Analysis of SAVEE-SS-App

We calculate the upper and lower bounds of the data rate as $\overline{r} = \max_{t^A \le t \le t^D} \min\{\max_{l \in L^{out}(v_s)} B^A_l(t), \max_{l \in L^{in}(v_d)} B^A_l(t)\}$ and $\underline{r} = \delta/(t^D - t^A)$. We also calculate the lower bound of the energy consumption and the upper bound of the power consumption on any path as $\overline{P} = \sum_{l' \in L'} P_{l'}(C_{l'})$ and $\underline{E} = \delta \overline{r}^{-1}(P^S_{s,0} + P^S_{d,0})$, respectively.

Theorem 5. *SAVEE-SS-App with* $\xi = (\epsilon - \epsilon_1)(1 + \epsilon_1)^{-1}\underline{r}^2\underline{E}\delta^{-1}\overline{P}^{-1}$ *finds a feasible triplet* (p, t_1, t_2) *of energy consumption within the least energy consumption multiplied by* $(1 + \epsilon)$ *in time* $O(T_A^2|L||D|(\log\log\log|D| + \epsilon_1^{-1})\delta\overline{P}\underline{E}^{-1}\underline{r}^{-2}(\overline{r} - \underline{r})(1 + \epsilon_1)(\epsilon - \epsilon_1)^{-1})$ *if the* ϵ_1*-approximation algorithm in [17] is used in Lines 10 and 16 in Alg. 2, where* $\epsilon_1 = -1 + \sqrt{1 + \epsilon}$.

Proof. Without loss of generality, we consider the energy consumption of data transfer starting at the beginning of the J-th time slot and ending at the K-th time slot on link l' as follows:

$$\begin{aligned} E_{l'}(r) =& \sum\nolimits_{j=J}^{K-1} (t[j] - t[j-1])(P_{l'}(r + r_{l'}[j]) - P_{l'}(r_{l'}[j])) \\ &+ [\delta r^{-1} - (t[K-1] - t[J-1])] \cdot (P_{l'}(r + r_{l'}[K]) \\ &- P_{l'}(r_{l'}[K])). \\ \frac{dE_{l'}(r)}{dr} =& \sum\nolimits_{j=J}^{K-1} (t[j] - t[j-1]) \frac{dP_{l'}(r + r_{l'}[j])}{dr} \\ &+ [\delta r^{-1} - (t[K-1] - t[J-1])] \frac{dP_{l'}(r + r_{l'}[K])}{dr} \\ &- \delta r^{-2}[P_{l'}(r + r_{l'}[K]) - P_{l'}(r_{l'}[K])] \\ &\ge -\delta r^{-2} P_{l'}(r + r_{l'}[K]) \ge -\delta r^{-2} P_{l'}(C_{l'}). \\ E_p(r) =& \sum\nolimits_{l' \in p} E_{l'}(r). \\ \frac{dE_p(r)}{dr} =& \sum_{l' \in p} \frac{dE_{l'}(r)}{dr} \ge -\delta r^{-2} \sum_{l' \in p} P_{l'}(C_{l'}) \ge -\delta r^{-2}\overline{P}. \end{aligned}$$

We use p^* and r^* to denote the path and the data rate of an optimal solution for data transfer from the J-th time slot to the K-th time slot, respectively. Let $\tilde{\xi} = r^* - B_0 - k\xi|_{B_0 + k\xi \le r^* < B_0 + (k+1)\xi, k \in Z_+}$ and $\hat{\xi} = \min\{\bar{\xi} \ge 0 | E_{p^*}(r^* - \bar{\xi}) = E_{p^*}(r^* - \tilde{\xi})\}$. We have $E_{p^*}(r^* - \hat{\xi}) \le E_{p^*}(r^*) - \hat{\xi} \min_{r^* - \hat{\xi} \le r \le r^*} \frac{dE_{p^*}(r)}{dr} \le E_{p^*}(r^*) + \hat{\xi}\delta r^{-2}\overline{P}$. Let $E(r)$ denote the energy consumption of the approximate solution obtained in Alg. 2 for data transfer from the J-th time slot to the K-th time slot. Let $E^*(r)$ denote the minimum energy consumption of data transfer from the J-th time slot to the K-th time slot

Algorithm 2: SAVEE-SS-App

Input: G, C_L, γ_L, R, ATB, $\{P^D_v(\cdot)\}$, $\{P^T_{v,i}(\cdot)\}$ and $\{P^R_{v,i}(\cdot)\}$
Output: Energy $E_{\min}$, path p, start time t_1, end time t_2

1: $E_{\min} = \infty$;
2: **for** $y = 1$ to T_A **do**
3: **for** $x = 0$ to $y - 1$ **do**
4: **for all** $l \in L$ **do**
5: $b_l = \min_{x \le i \le y-1} b_l[i]$;
6: $B_0 = \frac{\delta}{t[y] - t[x]}$; $B_1 = \frac{\delta}{t[y-1] - t[x]}$;
7: **for all** $\beta \in \{B_0 + k\xi | 0 \le k \le \lfloor (B_1 - B_0)/\xi \rfloor, k \in Z_+\}$ **do**
8: $\tau_1 = t[x]$; $\tau_2 = t[x] + \frac{\delta}{\beta}$;
9: Construct a directed graph $G'(V', L')$ from G with weights $w_{L'}$ and costs $c_{L'} = \{E_{l'}(\tau_1, \tau_2), l' \in L'\}$;
10: (ε, ρ) = the minimum energy cost and the corresponding path to transfer the data of size δ from v_s to v_d during time window $[\tau_1, \tau_2]$ on G';
11: **if** $\varepsilon < E_{\min}$ **then**
12: $E_{\min} = \varepsilon$; $p = \rho$; $t_1 = \tau_1$; $t_2 = \tau_2$;
13: $B_2 = \frac{\delta}{t[y] - t[x+1]}$;
14: **for all** $\beta \in \{B_0 + k\xi | 0 \le k \le \lfloor (B_2 - B_0)/\xi \rfloor, k \in Z_+\}$ **do**
15: $\tau_1 = t[y] - \frac{\delta}{\beta}$; $\tau_2 = t[y]$;
16: Compute the minimum energy consumption and the corresponding path (ε, ρ) in the same way as above, and update $E_{\min}$, p, t_1 and t_2 if needed;
17: **return** $(E_{\min}, p, t_1, t_2)$.

given the data rate r. Since $\xi = \frac{(\epsilon-\epsilon_1)\underline{r}^2\underline{E}}{(1+\epsilon_1)\delta\overline{P}}$,

$$\begin{aligned}\frac{E(r)}{E_{p^*}(r^*)} &\le \frac{(1+\epsilon_1)E^*(r)}{E_{p^*}(r^*)} \le \frac{(1+\epsilon_1)E_{p^*}(r)}{E_{p^*}(r^*)}\\ &\le (1+\epsilon_1)E_{p^*}(r^*-\hat{\xi})/E_{p^*}(r^*)\\ &\le (1+\epsilon_1)[E_{p^*}(r^*)+\hat{\xi}\delta r^{-2}\overline{P}]/E_{p^*}(r^*)\\ &= (1+\epsilon_1)[1+\frac{\hat{\xi}\delta\overline{P}}{r^2E_{p^*}(r^*)}]\\ &\le (1+\epsilon_1)[1+\frac{\hat{\xi}\delta\overline{P}}{r^2\underline{E}}] \le (1+\epsilon_1)[1+\frac{\xi\delta\overline{P}}{r^2\underline{E}}] = 1+\epsilon.\end{aligned}$$

The algorithm's execution time is calculated as $f(\epsilon_1) = \alpha_1 \cdot \frac{1+\epsilon_1}{\epsilon_1(\epsilon-\epsilon_1)}$. Let $\frac{df(\epsilon_1)}{d\epsilon_1} = \alpha_1 \cdot \frac{\epsilon_1^2+2\epsilon_1-\epsilon}{\epsilon_1^2(\epsilon-\epsilon_1)^2} = 0$. Then, $0 < \epsilon_1 = -1+\sqrt{1+\epsilon} < \epsilon$. Proof ends. □

Theorem 6. *SAVEE-SS-App with $\xi = \epsilon_1\underline{r}^2\underline{E}\delta^{-1}\overline{P}^{-1}$ finds a feasible triplet (p, t_1, t_2) of energy consumption within the least energy consumption multiplied by $(1+\epsilon_1)$ and PTFR within $(1+\epsilon_2)$ of requirement in time $O(T_A^2|D|(|L|+|D|\log|D|)\delta\overline{P}\underline{E}^{-1}\underline{r}^{-2}(\overline{r}-\underline{r})\cdot\epsilon_1^{-1}\epsilon_2^{-1})$ if the ϵ_2-approximation algorithm in [12] is used in Lines 10 and 16 in Alg. 2.*

Proof. The proof of the approximate ratio for energy consumption is similar to the proof of Theorem 5. Here, we derive the approximate ratio for the PTFR μ in the approximate solution. The ϵ_2-approximation algorithm in [12] guarantees that $-\log(1-\mu) \le [-\log(1-\gamma)](1+\epsilon_2)$. Let $f(\epsilon_2) = (1+\epsilon_2)\gamma + (1-\gamma)^{1+\epsilon_2}$. Since $f(0) = 1$ and when $\epsilon_2 \ge 0$, $df(\epsilon_2)/d\epsilon_2 = \gamma + (1-\gamma)^{1+\epsilon_2}\ln(1-\gamma) \ge \gamma - (1-\gamma)^{1+\epsilon_2}\cdot\gamma/(1-\gamma) = \gamma[1-(1-\gamma)^{\epsilon_2}] \ge 0$, we know that $f(\epsilon_2) \ge 1$. Hence, $\mu \le 1-(1-\gamma)^{1+\epsilon_2} \le (1+\epsilon_2)\gamma$. Proof ends. □

Lemma 2. *In SAVEE-SS-App, $dE_{l'}(r)/dr \ge -E_{l'}(r)/r$ if the data available time and the deadline of a data transfer request are in the same time slot in the ATB table.*

Proof. The energy consumption on link l' of data transfer in a single time slot is $E_{l'}(r) = \delta r^{-1}[P_{l'}(r+r_{l'}) - P_{l'}(r_{l'})]$. Since $r > 0$ and $dP_{l'}(r)/dr \ge 0$, we have $\frac{dE_{l'}(r)}{dr} = -\delta r^{-2}[P_{l'}(r+r_{l'}) - P_{l'}(r_{l'})] + \delta r^{-1}\frac{dP_{l'}(r+r_{l'})}{dr} = -\frac{E_{l'}(r)}{r} + \delta r^{-1}\frac{dP_{l'}(r+r_{l'})}{dr} \ge -\frac{E_{l'}(r)}{r}$. Proof ends. □

Theorem 7. *If the available time and the deadline for a data transfer request are in the same time slot in the ATB table, SAVEE-SS-App with $\xi = \underline{r}\cdot(\epsilon-\epsilon_1)/(1+2\epsilon-\epsilon_1)$ finds a feasible triplet (p, t_1, t_2) of energy consumption within the least energy consumption multiplied by $(1+\epsilon)$ in time $O(T_A^2|L||D|(\log\log\log|D|+\epsilon_1^{-1})\underline{r}^{-1}(\overline{r}-\underline{r})(1+2\epsilon-\epsilon_1)(\epsilon-\epsilon_1)^{-1})$ if the ϵ_1-approximation algorithm in [17] is used in Lines 10 and 16 in Alg. 2, where $\epsilon_1 = 1+2\epsilon-\sqrt{(1+\epsilon)(1+2\epsilon)}$.*

Proof. Here, p^*, r^*, $E(r)$, $E^*(r)$, $\tilde{\xi}$ and $\hat{\xi}$ are the same as defined in the proof of Theorem 5. Since $\hat{\xi} \le \xi = \underline{r}(\epsilon-\epsilon_1)/(1+2\epsilon-\epsilon_1) \le r^*(\epsilon-\epsilon_1)/(1+2\epsilon-\epsilon_1)$, $(r^*-\hat{\xi})(1+\epsilon_1) \le (r^*-2\hat{\xi})(1+\epsilon)$. Also, we have $E_{p^*}(r^*) \ge E_{p^*}(r^*-\hat{\xi}) + \hat{\xi}\cdot\min_{r^*-\hat{\xi}\le r\le r^*} dE_{p^*}(r)/dr \ge E_{p^*}(r^*-\hat{\xi}) - \hat{\xi}\cdot E_{p^*}(r^*-\hat{\xi})/(r^*-\hat{\xi}) = E_{p^*}(r^*-\tilde{\xi})\cdot(r^*-2\hat{\xi})/(r^*-\hat{\xi}) \ge E^*(r^*-\tilde{\xi})\cdot(r^*-2\hat{\xi})/(r^*-\hat{\xi})$. Hence, $\frac{E(r)}{E_{p^*}(r^*)} \le \frac{(1+\epsilon_1)E^*(r^*-\tilde{\xi})}{E_{p^*}(r^*)} \le \frac{(r^*-\hat{\xi})(1+\epsilon_1)E_{p^*}(r)}{(r^*-2\hat{\xi})E_{p^*}(r)} = \frac{(r^*-\hat{\xi})(1+\epsilon_1)}{r^*-2\hat{\xi}} \le 1+\epsilon$.

The algorithm's execution time is calculated as $f(\epsilon_1) = \alpha_2\cdot\frac{1+2\epsilon-\epsilon_1}{\epsilon_1(\epsilon-\epsilon_1)}$. Let $\frac{df(\epsilon_1)}{d\epsilon_1} = \alpha_2\cdot\frac{-\epsilon_1^2+2(1+2\epsilon)\epsilon_1-\epsilon(1+2\epsilon)}{\epsilon_1^2(\epsilon-\epsilon_1)^2} = 0$. Then, $0 < \epsilon_1 = 1+2\epsilon-\sqrt{(1+\epsilon)(1+2\epsilon)} < \epsilon$. Proof ends. □

Theorem 8. *If the data available time and the deadline for a data transfer request are in the same time slot in the ATB table, SAVEE-SS-App with $\xi = \underline{r}\cdot\epsilon_1/(1+2\epsilon_1)$ finds a feasible triplet (p, t_1, t_2) of energy consumption within the least energy consumption multiplied by $(1+\epsilon_1)$ and PTFR within $(1+\epsilon_2)$ of requirement in time $O(T_A^2|D|(|L|+|D|\log|D|)\underline{r}^{-1}(\overline{r}-\underline{r})\cdot(1+2\epsilon_1)\epsilon_1^{-1}\epsilon_2^{-1})$ if the ϵ_2-approximation algorithm in [12] is used in Lines 10 and 16 in Alg. 2.*

Proof. The proof is similar to Theorem 6 and Theorem 7, and hence is skipped. □

B. A Heuristic Algorithm for FPFB-MEC-SS

Considering the high time complexity of the proposed pseudo-polynomial approximation algorithm, we design a heuristic algorithm, named SAVEE-SS-Heu, for FPFB-MEC-SS, as shown in Alg. 3. Since T_A time slots form $T_A(T_A-1)/2$ time windows and the feasible data rate range can be divided into no more than $|L|$ data rate intervals, we only need to consider the subproblem in each time window and each data rate interval, and then the entire problem can be tackled by exhaustive search. For each subproblem, we solve it in three phases: 1) compute the average energy consumption on each link l' for data transfer of size δ over the data rate range (Lines 11, 21); 2) apply an existing algorithm to find a path with an approximately minimum energy consumption in a directed graph G' constructed from G (Lines 13, 21); 3) minimize the energy consumption of a univariate polynomial function with respect to data rate r over the data rate range to find the minimum energy cost and the corresponding data rate on a path obtained in Phase 2 (Lines 14, 21). If the time for Phase 2 is $T(\cdot)$, the time complexity of Alg. 3 is $O(T_A^2|L|T(\cdot))$.

For practical use, we make a tradeoff between objective optimality and time complexity, and thus use the ϵ-approximation algorithm in [12] to find a path with an approximately minimum energy consumption in Phase 2. To guarantee a feasible solution, we replace the user request $R(v_s, v_d, \delta, t^A, t^D, \gamma)$ with $R'(v_s, v_d, \delta, t^A, t^D, \gamma(1+\epsilon)^{-1})$ if there exists a feasible solution for R'. Otherwise, we only compute a feasible solution for R. In this case, the time complexity of Alg. 3 is $O(T_A^2|L||D|(|L|+|D|log|D|)\epsilon^{-1})$.

VII. Evaluation for Power-Down Model

A. Simulation Setup

SAVEE-PD-App is an instant bandwidth scheduling algorithm, which does not automatically guarantee the maximum overall energy saving in HPNs with continuously arriving user requests over a period of time. We conduct a simulation-based performance evaluation of SAVEE-PD-App in comparison with two algorithms: i) the minimum end time (MET) algorithm, which, for the maximum available bandwidth b of

Algorithm 3: SAVEE-SS-Heu

Input: G, C_L, γ_L, R, ATB, $\{P_v^D(\cdot)\}$, $\{P_{v,i}^T(\cdot)\}$ and $\{P_{v,i}^R(\cdot)\}$
Output: Energy $E_{\min}$, path p, start time t_1, end time t_2

1: $E_{\min} = \infty$;
2: **for** $y = 1$ to T_A **do**
3: **for** $x = 0$ to $y - 1$ **do**
4: **for all** $l \in L$ **do**
5: $b_l = \min_{x \le i \le y-1} b_l[i]$;
6: $B_0 = \frac{\delta}{t[y]-t[x]}$; $B_1 = \frac{\delta}{t[y-1]-t[x]}$;
7: $B = \{b_l | B_0 < b_l < B_1, l \in L\} \cup \{B_0, B_1\}$;
8: Sort $b_l \in B$ and label them as $b_1 < b_2 < \cdots < b_m$;
9: **for all** $b_i \in B - \{B_1\}$ **do**
10: Construct a directed graph $G'(V', L')$ with weights $w_{L'}$ from G;
11: Compute the average energy consumption $\bar{E}_{l'}$ on each link l' for data transfer with size δ over data rate range $[b_i, b_{i+1}]$ based on $\bar{E}_{l'} = \int_{b_i}^{b_{i+1}} E_{l'}(t[x], t[x] + \delta/r)dr \Big/ (b_{i+1} - b_i)$;
12: Assign $\bar{E}_{l'}$ to link l' as its cost $c_{l'}$;
13: Use the PDA algorithm in [12] to compute a path ρ with the minimum energy consumption in G';
14: Minimize a univariate polynomial $E_\rho = \sum_{l' \in \rho} E_{l'}(t[x], t[x] + \delta/r)$ in data rate range $[b_i, b_{i+1}]$ to find the minimum energy cost ε and the corresponding data rate r on a given path ρ;
15: **if** $\varepsilon < E_{\min}$ **then**
16: $E_{\min} = \varepsilon$; $p = \rho$; $t_1 = t[x]$; $t_2 = t[x] + \delta/r$;
17: $B_2 = \frac{\delta}{t[y]-t[x+1]}$;
18: $B = \{b_l | B_0 < b_l < B_2, l \in L\} \cup \{B_0, B_2\}$;
19: Sort $b_l \in B$ and label them as above;
20: **for all** $b_i \in B - \{B_2\}$ **do**
21: Compute the minimum energy cost ε, the corresponding path ρ and the corresponding data rate r in data rate range $[b_i, b_{i+1}]$ in the same way as above, and update $E_{\min}$, p, t_1 and t_2 if needed;
22: **return** $(E_{\min}, p, t_1, t_2)$.

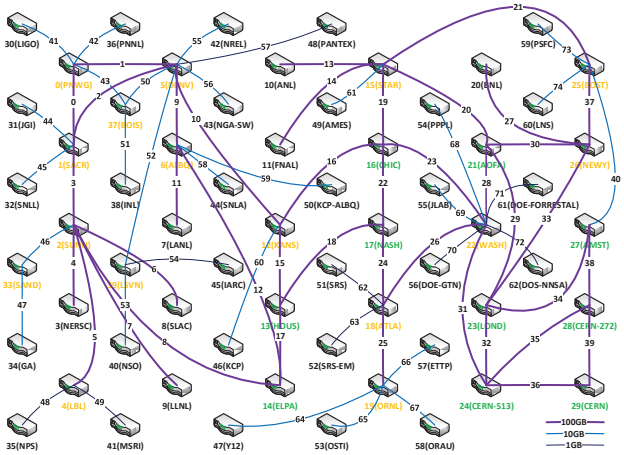


Fig. 8. The logical topology of ESnet [33] of 63 routers, where any of the ten trunk routers in green and edge routers in black could be shut down without affecting the network connectivity, while powering off any of the other 16 trunk routers in orange would break the network connectivity.

each link over each time-slot range $[i, j]$, calculates the path with the highest reliability and the available bandwidth of at least b in time-slot range $[i, j]$ to form a candidate solution set, and then selects the path and time-slot range of data transfer with the minimum end time meeting the reliability requirement from the candidate solution set, and ii) an energy-aware version of MET, referred to as EAMET. MET does not consider the energy consumption of network devices and always powers on all routers and line cards; while EAMET shuts down idle routers and line cards to achieve energy saving. In the simulation, the scheduler uses EAMET to find the earliest end time of data transfer, which is then used as a base point for setting an appropriate deadline constraint for SAVEE-PD-App in FPFB-MEC-PD.

We evaluate the performance of these algorithms in two types of networks: i) DOE's ESnet, which is a real-life HPN whose logical topology is shown in Fig. 8, and ii) semi-random simulated networks, where almost half of the routers form a grid, and each router in the rest connects to a randomly selected router in the grid. In semi-random simulated networks, we set the capacity of links in the grid to be 100 Gbps, and the capacity of other links to be 10 Gbps. The link error rate is randomly generated from a uniform distribution in [0, 0.01]. The arrivals of user requests follow a Poisson process, as widely adopted for traffic modeling in network simulation [24], [16], and hence the arrival interval of user requests follows an exponential distribution. We set the default average arrival interval of user requests to be one second and also consider multiple user requests that arrive in a burst within one second. The base-10 logarithm of the data sizes (in Gigabytes) follows a normal distribution $N(3.5, 0.5)$ such that 99.7% of the data sizes fall within a range from 0.8 Terabytes to 0.8 Petabytes. The difference between data available time and user request submission time in unit of seconds is randomly generated from an exponential distribution, whose average is set to be the data size divided by 10,000. The maximum PTFR in a user request is randomly generated from a uniform distribution in [0, 0.1]. Each simulation continuously processes 200 data transfer requests with different sources and destinations. Each simulation is repeated 10 times and each data point in a performance curve denotes the average result across 10 runs. The simulations in Section VIII use the same settings.

We set the parameters of the power model in Table II in reference to the specifications of Cisco CRS-3 100G router and Cisco 7603 10G router. According to their specifications, the maximum power consumptions of Cisco CRS (4-Slot) and CRS-3 (16-Slot) single-shelf systems are 3080W and 12320W, respectively, when the chassis is fully configured with line cards with traffic running [32], [31], and the power supply for Cisco 7603 Chassis is 950W [30]. In addition, [27] shows that the line card wakeup time has been reduced from 43s to 127ms based on the experimental results on an FPGA-based prototype. Nevertheless, the static and dynamic power consumption and booting-up/shutting-down time of different network devices could vary within a large range. Therefore, we use the normal distribution to model the variations in the parameter values of real network devices. The dynamic power model of each device is a linear function with respect to its instantaneous data transfer rate. We use the PDA algorithm in [12] with $\epsilon = 0.2$ in Lines 14 and 22 of SAVEE-PD-App. By default, we set the deadline of data transfer in SAVEE-PD-App to be 1.1 times of the minimum end time calculated by EAMET.

TABLE II
THE PARAMETERS OF THE POWER MODEL

Devices*	Stable SPC (W)	Booting-up SPC (W)	Shutting-down SPC (W)	Booting-up time (s)	Shutting-down time (s)	Max DPC / Max Data Rate (W/Gbps)
100G Routers	$N(1300, 65)$	$2\times$ Stable SPC	$2\times$ Stable SPC	$N(180, 9)$	$N(180, 9)$	$N(8, 2)$
10G Routers	$N(950, 50)$	$2\times$ Stable SPC	$2\times$ Stable SPC	$N(120, 6)$	$N(120, 6)$	$N(32, 8)$
1G Routers	$N(230, 15)$	$2\times$ Stable SPC	$2\times$ Stable SPC	$N(60, 3)$	$N(60, 3)$	$N(128, 32)$
100G Line cards	$N(450, 22)$	$2\times$ Stable SPC	$2\times$ Stable SPC	$N(30, 6)$	$N(30, 6)$	$N(2, 0.5)$
10G Line cards	$N(150, 8)$	$2\times$ Stable SPC	$2\times$ Stable SPC	$N(20, 4)$	$N(20, 4)$	$N(4, 1)$
1G Line cards	$N(70, 4)$	$2\times$ Stable SPC	$2\times$ Stable SPC	$N(10, 2)$	$N(10, 2)$	$N(6, 1.5)$

* A 100G/10G/1G router (or line card) is defined as a router (or line card) connected to a link with the capacity of at most 100/10/1 Gbps.

We define the incremental energy consumption (IEC) reduction (IECR) of SAVEE-PD-App over EAMET as:

$$IECR(\text{EAMET}) = \frac{IEC(\text{EAMET}) - IEC(\text{SAVEE-PD-App})}{IEC(\text{EAMET})} \cdot 100\%,$$

and the total energy consumption (TEC) reduction (TECR) of SAVEE-PD-App over another as:

$$TECR(\text{Other}) = \frac{TEC(\text{Other}) - TEC(\text{SAVEE-PD-App})}{TEC(\text{Other})} \cdot 100\%.$$

B. Traffic Load

We test MET, EAMET, and SAVEE-PD-App in ESnet under different traffic loads. We exponentially increase the average arrival interval of user requests within a range from 62.5 milliseconds to 16 seconds with a ratio of 2 to reflect a large variation in the dynamic distribution of traffic loads, in reference to the traffic parameter setting in [24], [16]. In the simulation, we run both EAMET and SAVEE-PD-App to schedule each incoming request. If the minimum end time and the corresponding PTFR calculated by EAMET meet the user requirement, we obtain the corresponding energy consumption for that request as the lower bound of the IEC, and then apply the schedule with less IEC produced by SAVEE-PD-App to the current user request; otherwise, there does not exist any feasible schedule.

We first evaluate the performance of SAVEE-PD-App in comparison with EAMET as an instant scheduling algorithm for FPFB-MEC-PD, and plot the average IECR measurements with the standard deviations in Fig. 9. These measurements show that SAVEE-PD-App saves IEC per user request from 10.8% to 12.4% over EAMET.

In practice, network service providers are always more concerned about the overall energy cost of the entire network than the individual energy cost of each request. Therefore, we evaluate the overall performance of MET, EAMET, and SAVEE-PD-App in terms of TEC of all user requests, and plot the average TECR measurements with the standard deviations in Fig. 9. These measurements show that SAVEE-PD-App saves energy by around 73% over MET and by 6.2% to 7.1% over EAMET under different average arrival intervals of user requests from 62.5 milliseconds to 16 seconds. Note that more frequent user requests result in higher traffic loads.

C. Deadline Constraints

We further examine the performance of MET, EAMET, and SAVEE-PD-App in ESnet under different data transfer deadline constraints. The data transfer deadline in SAVEE-PD-App is set to be from 1 to 1.3 times of the minimum end time calculated by EAMET, at an interval of 0.05. We plot the average IEC and TEC with the standard deviations in Fig. 10, which shows that SAVEE-PD-App saves IEC per user request by 7.4% to 16.1% over EAMET, and saves TEC by 72.8% to 73.9% over MET and by 3.3% to 7.4% over EAMET, as the deadline increases. We observe that the impact of deadline constraints on the energy-saving performance of SAVEE-PD-App is not very obvious, especially after the deadline constraint is extended to 1.1 times of the minimum transfer end time. These results provide a practical guidance for users to choose an appropriate deadline that is close to the minimum transfer end time calculated by EAMET.

D. Scalability

We run MET, EAMET, and SAVEE-PD-App in semi-random networks with different network sizes for scalability evaluation. The average IECR and TECR measurements with the standard deviations are plotted in Fig. 11, which shows that SAVEE-PD-App saves IEC per user request by 14.4% to 19.4% over EAMET, and saves TEC by 49.7% to 63.9% over MET and by 6.2% to 9.2% over EAMET, as the number of routers increases from 60 to 100 at an interval of 5.

E. Network Structures

We investigate MET, EAMET, and SAVEE-PD-App with various network structures, including a real-life topology (ESnet of 63 routers), semi-random topologies (a grid with random stars), a mesh topology (a grid with five routers per row), and purely random topologies. All networks except ESnet consist of 65 routers. We plot the average IECR and TECR measurements with the standard deviations in Fig. 12, which shows that SAVEE-PD-App saves IEC per user request by 11.9%, 18.7%, 18.5%, and 15.6% over EAMET, and saves TEC by 75.9%, 53.3%, 64.3%, and 60.4% over MET, and by 5.8%, 7.7%, 16.6%, and 11.7% over EAMET, in ESnet, semi-random, mesh, and purely random network topologies, respectively.

VIII. EVALUATION FOR SPEED-SCALING MODEL

A. Simulation Setup

We conduct a simulation-based performance evaluation of SAVEE-SS-Heu in comparison with two algorithms: MET and a linear power function-based algorithm, referred to as LPF-SS, which is used for comparison to evaluate the dynamic energy saving achieved by polynomial power functions. LPF-SS takes the following procedure: i) evenly selects 1000 samples from the polynomial power function of each device within its data rate range and generates a linear power function using the least-squares fitting; ii) uses the PDA algorithm in [12] to search for the approximately minimum DEC within each time-slot range based on generated linear power functions. Similarly, in the simulation, the scheduler uses MET to find

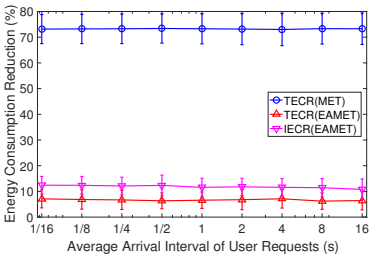


Fig. 9. The energy saving of SAVEE-PD-App with different arrival intervals of user requests in ESnet.

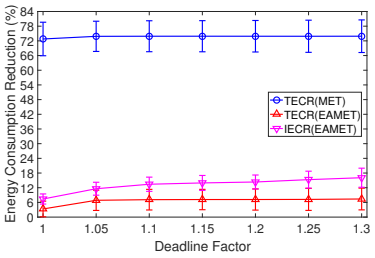


Fig. 10. The energy saving of SAVEE-PD-App with different deadline constraints in ESnet.

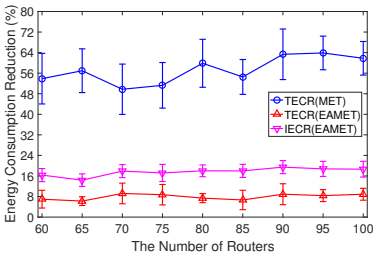


Fig. 11. The energy saving of SAVEE-PD-App under different network sizes in semi-random networks.

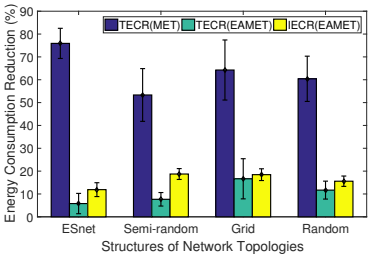


Fig. 12. The energy saving of SAVEE-PD-App with different network structures.

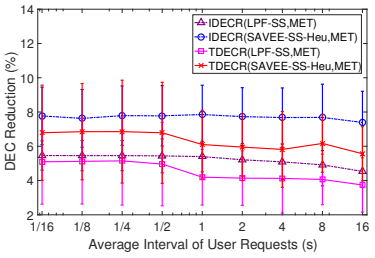


Fig. 13. The dynamic energy saving of SAVEE-SS-Heu and LPF-SS over MET with different arrival intervals of user requests in ESnet.

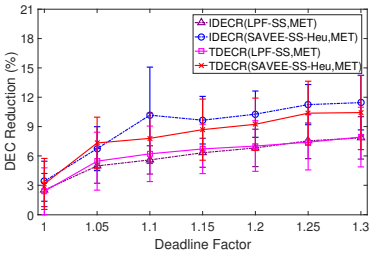


Fig. 14. The dynamic energy saving of SAVEE-SS-Heu and LPF-SS over MET with different deadline constraints in ESnet.

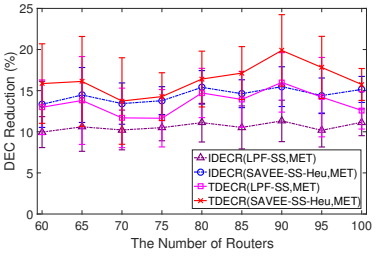


Fig. 15. The dynamic energy saving of SAVEE-SS-Heu and LPF-SS over MET under different network sizes in semi-random networks.

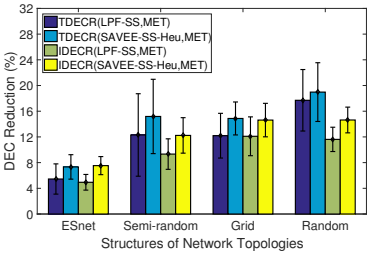


Fig. 16. The dynamic energy saving of SAVEE-SS-Heu and LPF-SS over MET with different network structures.

TABLE III
TEMPLATES FOR THE POLYNOMIAL FUNCTIONS OF DPC
$(p'(r) \geq 0, r \in [0,1])$

Linear	$p = r$.
Concave	$p = -r^2 + 2r$, $p = r^3 - 3r^2 + 3r$, $p = -r^4 + 4r^3 - 6r^2 + 4r$.
Convex	$p = r^2$, $p = r^3$, $p = r^4$.
Non-convex and Non-concave	$p = 2.2857r^3 - 1.7143r^2 + 0.4286r$, $p = 2.2857r^3 - 5.1429r^2 + 3.8571r$, $p = 4r^3 - 6r^2 + 3r$, $p = 2r^4 - 4r^3 + 3r^2$, $p = 0.3333r^4 - 1.3333r^3 + 2r^2$, $p = 31.6098r^5 - 79.0244r^4 + 67.3171r^3 - 21.9512r^2 + 3.0488r$, $p = 33.3913r^5 - 83.4783r^4 + 76.5217r^3 - 31.3043r^2 + 5.8696r$.

the earliest end time of data transfer, which is then used as a base point for setting an appropriate deadline constraint for LPF-SS and SAVEE-SS-Heu in FPFB-MEC-SS.

The dynamic power model of each device is a polynomial function with respect to its instantaneous data transfer rate, and is randomly selected from a set of polynomial function templates with both the data rate and DPC normalized to the range of $[0, 1]$, as listed in Table III. The maximum data rate of a router is the sum of the capacities of all incoming and outgoing links, and the maximum data rate of the transmitter (or the receiver) of a line card is the sum of the capacities of all outgoing (or incoming) links. We follow the parameter values in the last column of Table II to set the maximum DPC and adapt the selected polynomial function templates to the corresponding ranges of data rate and DPC. By default, the deadline of data transfer in SAVEE-SS-Heu and LPF-SS is set to be 1.1 times the minimum end time calculated by MET. Other settings remain the same as in Subsection VII-A.

We define the incremental dynamic energy consumption (IDEC) reduction (IDECR) of an algorithm over another as:

$$IDECR(\text{Alg. } 1, \text{Alg. } 2) = \frac{IDEC(\text{Alg. } 2) - IDEC(\text{Alg. } 1)}{IDEC(\text{Alg. } 2)} \cdot 100\%,$$

and the total dynamic energy consumption (TDEC) reduction (TDECR) of an algorithm over another as:

$$TDECR(\text{Alg. } 1, \text{Alg. } 2) = \frac{TDEC(\text{Alg. } 2) - TDEC(\text{Alg. } 1)}{TDEC(\text{Alg. } 2)} \cdot 100\%.$$

B. Traffic Load

We examine the performance of MET, LPF-SS, and SAVEE-SS-Heu in ESnet under different traffic loads. The average arrival interval of user requests exponentially increases from 62.5 milliseconds to 16 seconds with a ratio of 2. We plot the average IDECR and TDECR measurements of LPF-SS and SAVEE-SS-Heu over MET with the standard deviations

in Fig. 13, which illustrates that LPF-SS and SAVEE-SS-Heu reduce IDEC per user request by 4.5% to 5.4% and by 7.4% to 7.9%, and TDEC by 3.7% to 5.1% and by 5.6% to 6.9%, in comparison with MET, respectively. These performance curves clearly show that SAVEE-SS-Heu based on polynomial power functions saves more DEC than LPF-SS based on linear power functions.

C. Deadline Constraints

We investigate the performance of MET, LPF-SS, and SAVEE-SS-Heu in ESnet under different data transfer deadline constraints. The data transfer deadline in SAVEE-SS-Heu and LPF-SS is set to be from 1 to 1.3 times the minimum end time calculated by MET, at an interval of 0.05. We plot the average IDECR and TDECR of LPF-SS and SAVEE-SS-Heu over MET with the standard deviations in Fig. 14, which show that LPF-SS and SAVEE-SS-Heu reduces IDEC per user request by 2.5% to 7.8% and by 3.4% to 11.5%, and TDEC by 2.4% to 7.9% and by 3.1% to 10.4%, in comparison with MET, respectively, as the transfer deadline increases. It is also interesting to point out that the impact of deadline constraints on the dynamic energy saving of SAVEE-SS-Heu becomes less significant when it is larger than 1.1 times the minimum end time. Similarly, these results provide a practical guidance for users to choose an appropriate deadline constraint, which is around 1.1 times the minimum transfer end time calculated by MET.

D. Scalability

We run MET, LPF-SS, and SAVEE-SS-Heu in semi-random networks with different network sizes for scalability evaluation. The number of routers in the networks increases from 60 to 100 at an interval of 5. We plot the average IDECR and TDECR of LPF-SS and SAVEE-SS-Heu over MET with the standard deviations in Fig. 15. These measurements show that LPF-SS and SAVEE-SS-Heu reduce IDEC per user request by 10.0% to 11.3% and by 13.3% to 15.5%, and TDEC by 11.6% to 16.0% and by 13.7% to 19.9%, in comparison with MET, respectively.

E. Network Structures

We evaluate MET, LPF-SS, and SAVEE-SS-Heu under various network structures. The average IDECR and TDECR of LPF-SS and SAVEE-SS-Heu over MET with the standard deviations are plotted in Fig. 16. These measurements show that LPF-SS and SAVEE-SS-Heu reduce TDEC by 5.5% and 7.3%, by 12.3% and 15.2%, by 12.2% and 14.9%, as well as by 17.7% and 19.0% over MET in real-life, semi-random, mesh, and purely random network topologies, respectively.

IX. Conclusion

We formulated two advance instant bandwidth scheduling problems FPFB-MEC-PD and FPFB-MEC-SS in HPNs using two power models, i.e. power-down and speed-scaling, for minimizing energy consumption of data transfer under deadline and reliability constraints. Both of the problems have been proved to be NP-complete. We designed an FPTAS for FPFB-MEC-PD, and designed an approximation algorithm and a heuristic algorithm for FPFB-MEC-SS.

Our work reveals that energy-aware bandwidth scheduling could lead to significant energy saving in comparison with the existing scheduling algorithms with focus on traditional optimization objectives. It is of our future interest to incorporate the proposed bandwidth scheduling algorithms into the control plane of existing high-performance networks such as ESnet OSCARS, and evaluate the energy saving performance in real-life network environments.

Acknowledgment

This research is sponsored by U.S. Department of Energy's Office of Science under Grant No. DE-SC0015892 and National Science Foundation under Grant No. CNS-1560698 with New Jersey Institute of Technology. We would also like to thank three anonymous reviewers for their constructive comments that have greatly helped us improve the technical quality and presentation of this manuscript.

References

[1] OSCARS: On-demand Secure Circuits and Advance Reservation System. http://www.es.net/oscars.

[2] Internet2 Interoperable On-Demand Network (ION) Service. http://www.internet2.edu/ion.

[3] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sanso, "Energy management through optimized routing and device powering for greener communication networks," *IEEE/ACM Tran. on Net.*, vol. 22, no. 1, pp. 313–325, 2014.

[4] M. Andrews, A.F., L. Zhang, and W. Zhao, "Routing for power minimization in the speed scaling model," *IEEE/ACM Tran. on Net.*, vol. 20, no. 1, pp. 285–294, 2012.

[5] M. Andrews, A. Anta, L. Zhang, and W. Zhao, "Routing and scheduling for energy and delay minimization in the powerdown model," *Wiley Networks*, vol. 61, no. 3, pp. 226–237, 2013.

[6] M. Andrews, S. Antonakopoulos, and L. Zhang, "Energy-aware scheduling algorithms for network stability," in *Proc. of IEEE INFOCOM*, Shanghai, China, Apr 2011, pp. 1359–1367.

[7] A. Antoniadis, S. Im, R. Krishnaswamy, B. Moseley, V. Nagarajan, K. Pruhs, and C. Stein, "Hallucination helps: Energy efficient virtual circuit routing," in *Proc. of ACM-SIAM SODA*, Portland, Oregon, USA, Jan 2014, pp. 1141–1153.

[8] A. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A survey of green networking research," *IEEE Comm. Surveys and Tutorials*, vol. 14, no. 1, pp. 3–20, 2012.

[9] K. Bilal, S. Khan, S. Madani, K. Hayat, M. Khan, N. Min-Allah, J. Kolodziej, L. Wang, S. Zeadally, and D. Chen, "A survey on green communications using adaptive link rate," *Cluster Comp.*, vol. 16, no. 3, pp. 575–589, 2013.

[10] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli, "Green networking with packet processing engines: Modeling and optimization," *IEEE/ACM Tran. on Net.*, vol. 22, no. 1, pp. 110–123, 2014.

[11] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proc. of IEEE INFOCOM*, Phoenix, AZ, USA, Apr 2008, pp. 457–465.

[12] S. Chen, M. Song, and S. Sahni, "Two techniques for fast computation of constrained shortest paths," *IEEE/ACM Tran. on Net.*, vol. 16, no. 1, pp. 105–115, 2008.

[13] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP network energy cost: Formulation and solutions," *IEEE/ACM Tran. on Net.*, vol. 20, no. 2, pp. 463–376, 2012.

[14] Cisco. (2009, Jan) Miercom lab testing summary report: Aggregation services router - power efficient. [Online]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/ routers/asr-1000-series-aggregation-services-routers/ asr1000_series_green.pdf

[15] ——. (2010, Jan) Miercom lab testing summary report: Integrated services routers g2. [Online]. Available: http://miercom.com/pdf/reports/20100110.pdf

[16] R. Cohen, N. Fazlollahi, and D. Starobinski, "Graded channel reservation with path switching in ultra high capacity networks," in *IEEE International Conference on Broadband Communications, Networks and Systems (BROADNETS)*, San Jose, CA, USA, Oct. 2006.

[17] G. Feng and T. Korkmaz, "A fast hybrid ϵ-approximation algorithm for computing constrained shortest paths," *IEEE Comm. Letters*, vol. 17, no. 7, pp. 1471–1474, 2013.

[18] M. Garey and D. Johnson, Eds., *A List of NP-Complete Problems*. Bell Telephone Laboratories, 1979.

[19] A. Gupta, R. Krishnaswamy, and K. Pruhs, *Online Primal-Dual for Non-linear Optimization with Applications to Speed Scaling*, T. Erlebach and G. Persiano, Eds. Springer Berlin Heidelberg, 2013.
[20] M. Gupta and S. Singh, "Greening of the Internet," in *Proc. of ACM SIGCOMM*, Karlsruhe, Germany, Aug 2003, pp. 19–26.
[21] R. Krishnaswamy, V. Nagarajan, K. Pruhs, and C. Stein, "Cluster before you hallucinate: Approximating node-capacitated network design and energy efficient routing," in *Proc. of ACM STOC*, New York, NY, USA, Jun 2014, pp. 734–743.
[22] D. Li, Y. Shang, and C. Chen, "Software defined green data center network with exclusive routing," in *Proc. of IEEE INFOCOM*, Toronto, Canada, May 2014, pp. 1743–1751.
[23] Q. Li, M. Xu, Y. Yang, L. Gao, Y. Cui, and J. Wu, "Safe and practical energy-efficient detour routing in ip networks," *IEEE/ACM Tran. on Net.*, vol. 22, no. 6, pp. 1925–1937, 2014.
[24] Y. Lin and Q. Wu, "Complexity analysis and algorithm design for advance bandwidth scheduling in dedicated networks," *ACM/IEEE Trans. on Net.*, vol. 21, no. 1, pp. 14–27, 2013.
[25] F. Moghaddam, P. Lago, and P. Grosso, "Energy-efficient networking solutions in cloud-based environments: A systematic literature review," *ACM Comp. Surveys*, vol. 47, no. 4, pp. 64:1–64:32, 2015.
[26] A. Orgerie and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Comp. Surveys*, vol. 46, no. 4, pp. 47:1–47:31, 2014.
[27] T. Pan, T. Zhang, J. Shi, Y. Li, L. Jin, F. Li, J. Yang, B. Zhang, X. Yang, M. Zhang, H. Dai, and B. Liu, "Towards zero-time wakeup of line cards in power-aware routers," *IEEE/ACM Tran. on Net.*, vol. 24, no. 3, pp. 1448–1461, 2016.
[28] T. Shu, C. Wu, and D. Yun, "Advance bandwidth reservation for energy efficiency in high-performance networks," in *Proc. of IEEE LCN*, Sydney, Australia, Oct 2013, pp. 541–548.
[29] J. Tang, B. Mumey, Y. Xing, and A. Johnson, "On exploiting flow allocation with rate adaptation for green networking," in *Proc. of IEEE INFOCOM*, Orlando, Florida, USA, Mar 2012, pp. 1683–1691.
[30] Cisco 7603 Chassis, 2016, http://www.cisco.com/c/en/us/products/collateral/routers/7603-router/product_data_sheet09186a0080088771.pdf.
[31] Cisco CRS-3 16-Slot Single-Shelf System, 2016, https://www.spectra.com/wp-content/uploads/CRS-3_16-Slot.pdf.
[32] Cisco CRS 4-Slot Single-Shelf System, 2013, http://www.cisco.com/c/en/us/products/collateral/routers/carrier-routing-system/CRS-3_4-Slot_DS.html.
[33] Lawrence Berkeley National Laboratory, U.S. Department of Energy. (2016) Energy sciences network (ESnet). [Online]. Available: https://my.es.net/
[34] OSCARS: How It Works, 2016, https://www.es.net/engineering-services/oscars/how-it-works/.
[35] A. Vishwanath, K. Hinton, R. Ayre, and R. Tucker, "Modeling energy consumption in high-capacity routers and switches," *IEEE J. on Selected Areas in Comm.*, vol. 32, no. 8, pp. 1524–1532, 2014.
[36] L. Wang, F. Zhang, K. Zheng, A. Vasilakos, S. Ren, and Z. Liu, "Energy-efficient flow scheduling and routing with hard deadlines in data center networks," in *Proc. of IEEE ICDCS*, Madrid, Spain, Jul 2014, pp. 248–257.
[37] X. Wang, Y. Yao, X. Wang, and Q. Cao, "CARPO: Correlation-aware power optimization in data center networks," in *Proc. of IEEE INFOCOM*, Orlando, FL, USA, Mar 2012, pp. 1125–1133.
[38] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering," in *Proc. of IEEE ICNP*, Kyoto, Japan, Oct 2010, pp. 21–30.
[39] Z. Zhang, Y. Bejerano, and S. Antonakopoulos, "Energy-efficient IP core network configuration under general traffic demands," *IEEE/ACM Tran. on Net.*, vol. 24, no. 2, pp. 745–758, 2016.
[40] Y. Zhao, S. Wang, S. Xu, X. Wang, X. Gao, and C. Qiao, "Load balance vs energy efficiency in traffic engineering: A game theoretical perspective," in *Proc. of IEEE INFOCOM*, Turin, Italy, Apr 2013, pp. 530–534.

Tong Shu received the B.S. degree in information management and system from Peking University, P.R. China in 2005 and the M.S. degree in Computer Science from the University of Memphis in 2015. She is currently a doctoral student in the Department of Computer Science at New Jersey Institute of Technology, and works in the Big Data Group. Her research interests include big data, green networking and computing, cloud computing, and computer networks.

Chase Q. Wu received the Ph.D. degree in computer science from Louisiana State University in 2003. He was a research fellow at Oak Ridge National Laboratory during 2003-2006 and an assistant and associate professor at University of Memphis during 2006-2015. He is currently an associate professor with the Department of Computer Science at New Jersey Institute of Technology. His research interests include big data, distributed and parallel computing, and computer networks.