Energy-Efficient, Delay-aware Packet Scheduling in High-Speed Networks

Qun Yu*, Taieb Znati† and Wang Yang‡
*† University of Pittsburgh, Pittsburgh, PA 15260
Email: {quy3, znati}@pitt.edu
‡Southeast University, Nanjing, China 210096
Email: wyang@njnet.edu.cn

***Abstract*—In current commercial routers, increased execution speeds of Network Processor Units (NPUs) in Line Cards (LCs) significantly improve network QoS performance. Achieving high network performance, however, may come at a high cost of routers' energy consumption. Dynamic Voltage Frequency Scaling (DVFS) and Dynamic Power Management (DPM) have been proposed as schemes to manage power and reduce energy consumption. Excessive reduction in execution rates or extended sleep periods to save energy, however, could result in severe network degradation which in turn may lead to violation of QoS requirements of the underlying applications. To address the energy-QoS dichotomy, we propose a congestion- and energy-aware packet scheduling scheme to achieve a balance between network delay and energy saving. The scheme, referred to as Queue Length (QL)-based Delay-aware packet scheduler (QLDA), uses multiple queue length thresholds to accurately capture network congestion. In response to different levels of network congestion, the QLDA scheme uses carefully designed frequency adjustment strategies to control execution rates in line cards and achieve high energy savings, without violating the delay requirements of the underlying applications. The simulation results show that the QLDA scheme has potential for significant energy saving in high-speed networks. Furthermore, a simulation study is carried out to compare the performance of the proposed scheme to other DVFS-based schemes described in the literature. The results show that the QLDA scheme outperforms the existing schemes for different network topologies and traffic loads, while meeting the delay performance of the supported applications.**

keywords. *Energy-efficient, Delay-aware packet scheduling, DVFS, DPM, Network performance, Simulation.*

I. Introduction

The potential environmental impact of high energy consumption, coupled with the rising cost of computing and networking infrastructure, has become a major concern in an increasingly IT-reliant society [1]. Today's high-performance router LCs handle most traffic processing, buffering and forwarding, using specialized ASIC or other programmable hardware [2]. It has been reported that LCs consume about 70% of the total router power, with the NPUs consuming more than 50% of the power consumed by one LC [3], [4]. Recent advances in semiconductor technology, which enabled higher parallelism and increased clock frequencies, paved the way to a new generation of power routers. These advances, however, come at a heavy price of increased power consumption, due to higher line card speed [2]. Therefore, seeking efficient solutions to reducing power consumption, without adversely affecting network performance, becomes a critical design objective of future networks.

Currently, two approaches are frequently used to manage power in computing and networking environments. The first, referred to as *Speed Scaling*, uses Dynamic Voltage and Frequency Scaling (DVFS) to control execution rates and reduce power and energy consumption [5]–[10]. The second, referred to as *Dynamic Power Management (DPM)*, uses sleep mode to control power [7], [8]. A large body of research work shows that DVFS can provide the basis for viable solutions to achieve significant power savings in computing, communications and storage devices [5]–[10]. However, excessively slowing down of the processors may lead to unacceptable level of QoS degradation of the supported applications. Dynamically adjusting processors' execution rates to achieve energy saving, while satisfying QoS requirements, remains a challenge.

To address this challenge, we propose a novel QL-based, Delay-aware, DVFS-enabled packet scheduler, referred to as QLDA, to reduce energy consumption, while adhering to the QoS requirements of the supported applications. QLDA uses multiple queue length thresholds to model network congestion, and the exponentially weighted moving average (EWMA) algorithm to predict average queue length and average packet delay. In response to different levels of network congestion, different NPU-rate scaling strategies then are used to determine when and how NPU execution rates are adjusted based on the predicted queue-length and packet-delay. The goal is to achieve high energy savings, without degrading delay performance. A simulation study, based on a comprehensive energy model, is used to investigate the performance of the proposed packet scheduler, in different networking environments and traffic loads. The results show that QLDA achieves significant energy saving. Furthermore, the results of a comparative analysis study shows that QLDA outperforms similar QoS-aware DVFS-enabled schemes while maintaining the acceptable QoS requirements, with low scheduling overhead.

The rest of this paper is organized as follows: the related work is discussed in Section II. A Delay-aware DVFS-enabled scheduler and its workload prediction mechanisms and scaling strategies are presented in Section III. An energy model is introduced in Section IV. The performance of the proposed scheme is discussed, and compared with other schemes, under different network environments and traffic loads in Section V.

978-1-4673-8590-9/15/$31.00 ©2015 IEEE

Section VI presents the conclusion of this paper.

II. Related Work

Several energy-efficient schemes have been proposed for green networks [5], [6], [10]. Some of these schemes propose energy-based traffic engineering approaches designed to only keep a sufficient number of active routers, linecards and interfaces to support the network workload. The remaining network devices are either shutdown or put into sleep mode. Other research works focus on energy saving using DVFS-based power management approaches. In [7], Nedevschi, et al. present two simple power management algorithms, and explore the effect of sleep mode and DVFS-based rate adaptation on network energy saving. In [9], Mandviwalla et al. propose three load-dependent strategies, i.e. Value Predictor (VP), Moving Average Predictor (MAP) and Exponentially-Weighted MAP (EWMAP), to reduce energy consumption in multiprocessor-based LCs. The results show that more than 60% dynamic power savings of the maximal dynamic power consumption can be achieved in one LC. Although these proposed schemes seek to reduce dynamic energy consumption at different levels through using link utilization in DVFS-enabled processors, they do not address the impact of the entire energy savings on QoS performance under different traffic loads in a network. On the other hand, considering the lack of a comprehensive router-based energy model, in [11], Vishwanath, Arun, et al. propose a power model measurement methodology that quantifies the energy efficiency of high-capacity routing platforms at the packet- and byte-level. It focuses on network energy evaluation. The approach used to save energy, however, is not discussed and analyzed in detail. In this paper, the proposed Delay-aware DVFS-enable packet scheduler addresses these shortcomings, and seeks a balanced tradeoff between high network energy savings and acceptable levels of network QoS performance under different traffic loads, based on a derived comprehensive router-based energy model. This is achieved by controlling the NPU execution rates based on **queue length**.

III. Delay-aware DVFS-Scheduler

In this section, we first present the basic Delay-aware, DVFS-enabled scheduling architecture. We then discuss QLDA, including the frequency scaling strategies it uses to adjust the NPU's execution rates based on queue length.

A. Delay-aware DVFS-Scheduler Design and Architecture

The basic idea of DVFS-Scheduler is to dynamically adjust the processor frequency, based on the current state of the network, to reduce energy consumption. To design an effective QL-based Delay-aware DVFS-Scheduler, several issues must be addressed. First, a strategy must be in place to determine how queue length impacts scheduling decisions. Second, appropriate levels of congestion granularity must be taken into consideration when adjusting the NPU's execution rate. Multiple queue length thresholds to model different levels of network congestion are considered in this paper. Third, a mechanism must be in place to predict traffic end-to-end delay and bind its variability so that the desired delay performance can be achieved. In the proposed scheduler, an effective method, which estimates packet delay variability to predict the deviation of packet delay from the target end-to-end delay, is used to control NPU's execution rate adjustments. Finally, an adaptive mechanism must be in place to control the "aggressivity" of the scheduling policy to ensure energy savings without degrading QoS performance.

To address the above issues, a Delay-aware DVFS-enabled scheduling architecture, depicted in Fig. 1, is proposed. The Traffic Monitor (TM) monitors the packet queue, and gathers statics related to its length. The estimated average queue length, $\bar{q}(\tau)$, and average packet delay, $\bar{d}(\tau)$, over a time interval τ, are used to scale up or down the NPU execution rates. The NPU Rate Scaler (RS) computes a network state-dependent scaling function, $\xi()$, taking into consideration the aggressivity factor of the scheduling strategy, $\eta(\tau)$ generated by the average network traffic load $\rho(\tau)$, and the current level of network congestion. The DVFS Adjustor (DA) adjusts the NPU frequency, $f(\tau)$, based on the scaling factor.

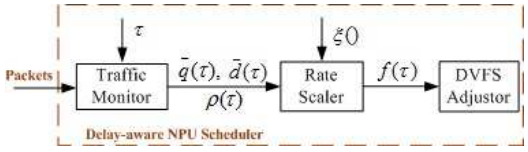

Fig. 1: Delay-aware DVFS-enabled Scheduling Architecture

Based on the above architecture, a Delay-aware DVFS-enabled packet scheduler is proposed, which uses predicted average queue length and average packet delay to control the frequency adjustment. The related NPU rate scaling strategies, queue-length and packet-delay prediction mechanisms are introduced in the following.

B. QL-based Delay-Aware Packet Scheduler (QLDA)

QLDA uses the exponentially weighted moving average (EWMA) scheme to periodically predict the average queue length over a given time interval, τ, to adjust the NPU execution frequency dynamically. In order to reduce DVFS switching overhead, QLDA uses a coarser level of network congestion granularity in its decision to scale up or down the NPU execution rates. More specifically, QLDA uses two queue length thresholds, namely q_l and q_h ($0 \leq q_l < q_h \leq Q$) to define ***low, medium*** and ***high*** network congestion regions, as depicted in Fig.2.

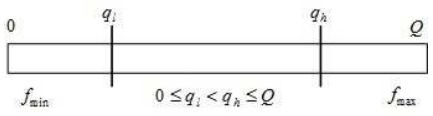

Fig. 2: Packet buffer.

Based on the above three packet buffer occupancy regions, the frequency, f_{τ_k}, over the k^{th} time interval, τ_k, $k \geq 1$, is defined in Eq. 1.

$$f_{\tau_k} = \begin{cases} f_{min}, & \text{if } \overline{q}_{\tau_k} \leq q_l \\ f_{max}, & \text{if } \overline{q}_{\tau_k} \geq q_h \\ f_{\tau_{k-1}}, & \text{if } q_l < \overline{q}_{\tau_k} < q_h \text{ and } \left|\overline{d}_{\tau_k} - d^T\right| \leq \tilde{dv}_{\tau_k} \\ f_{min} + (f_{max} - f_{min}) \cdot \xi_{\tau_k}, & \\ & \text{if } q_l < \overline{q}_{\tau_k} < q_h \text{ and } \left|\overline{d}_{\tau_k} - d^T\right| > \tilde{dv}_{\tau_k} \end{cases} \quad (1)$$

In the above strategy function, the scaling factor, ξ_{τ_k}, over τ_k, as illustrated in Eq. 2, is determined based on the queue occupancy in the middle region (q_l, q_h), defined as the ratio of the queue length occupancy to the difference between two queue length thresholds, raised to the power, $\eta(\rho)$.

$$\xi_{\tau_k} = (\frac{\overline{q}_{\tau_k} - q_l}{q_h - q_l})^{\eta(\rho)} \quad (2)$$

The average queue length, $\overline{q}_{\tau_k}$, over τ_k, is defined in Eq. 3. q_{τ_k} represents the queue length at the end of the interval τ_k, and $w^q_{\tau_k}$, defined as $w^q_{\tau_k} = c_q \cdot \frac{eq^2_{\tau_k}}{\sigma q_{\tau_k}}$, where $0 < c_q < 1$, $0 < w^q_{\tau_k} < 1$, is the queue-length weight factor.

$$\overline{q}_{\tau_k} = (1 - w^q_{\tau_k}) \cdot \overline{q}_{\tau_{k-1}} + w^q_{\tau_k} \cdot q_{\tau_k} \quad (3)$$

The term eq_{τ_k} represents the queue length prediction error function, defined as $eq_{\tau_k} = q_{\tau_k} - \overline{q}_{\tau_k}$, and σq_{τ_k} denotes the square prediction error for τ_k, defined as $\sigma q_{\tau_k} = c_q \cdot eq^2_{\tau_k} + (1 - c_q) \cdot \sigma q_{\tau_{k-1}}$. The first order auto-regressive filter used to predict future queue length, combined with the error prediction method used to adaptively compute the weight function, $w^q_{\tau_k}$, guarantee that the predicted queue length is not affected by small deviations.

The aggressivity factor $\eta(\rho_{\tau_k})$ associated with the average traffic load ρ_{τ_k}, over τ_k, is defined as Eq. 4.

$$\eta(\rho_{\tau_k}) = a \cdot e^{-(\frac{\rho_{\tau_k} - b}{c})^2} \quad (4)$$

The aggressivity function, $\eta()$, uses Gaussian regression model to generate the aggressivity factor of the scheduling strategy based on the traffic load. a, b, and c are constant model parameters. The average traffic load ρ_{τ_k} is computed by the traffic average arrival rate $\overline{\lambda}_{\tau_k}$ over τ_k and the maximal NPU service rate u_{max}, as illustrated Eq. 5.

$$\rho_{\tau_k} = \frac{\overline{\lambda}_{\tau_k}}{u_{max}} \quad (5)$$

In order to further reduce DVFS scaling overhead, QLDA uses the estimated delay variance to decide when to adjust NPU's frequency. When the estimated average queue length falls in the middle region (q_l, q_h), QLDA does not systematically scale up or down the NPU's execution rate over every time interval τ. DVFS scaling in this region only takes place when the absolute value of the deviation between the predicted average packet delay, $\overline{d}_{\tau_k}$, over τ_k, and the target packet delay, d^T, exceeds the estimated delay deviation, $\tilde{dv}_{\tau_k}$. When deviation occurs, the frequency is scaled up or down, depending on the queue length and the target packet delay, as illustrated in Eq. 1.

In order to predict the average packet delay, $\overline{d}_{\tau_k}$, the same exponential smoothing technique, defined in Eq. 6, is used. d_{τ_k} represents the k^{th} average packet delay, which is computed based on the average queue length, the average arrival rate and the average departure rate over τ_k.

$$\overline{d}_{\tau_k} = (1 - w^d_{\tau_k}) \cdot \overline{d}_{\tau_{k-1}} + w^d_{\tau_k} \cdot d_{\tau_k} \quad (6)$$

The term $w^d_{\tau_k}$, $0 < w^d_{\tau_k} < 1$, denotes a dynamic delay weight factor and is defined as $w^d_{\tau_k} = c_d \cdot \frac{ed^2_{\tau_k}}{\sigma d_{\tau_k}}$, where $0 < c_d < 1$. Similarly, ed_{τ_k} represents the delay prediction error function, defined as $ed_{\tau_k} = d_{\tau_k} - \overline{d}_{\tau_k}$, and σd_{τ_k} denotes the square prediction error for τ_k, defined as $\sigma d_{\tau_k} = c_d \cdot ed^2_{\tau_k} + (1 - c_d) \cdot \sigma d_{\tau_{k-1}}$. The first order auto-regressive filter used to predict future packet delay, combined with the error prediction method used to adaptively compute the weight function $w^d_{\tau_k}$, guarantee that the predicted delay is not affected by small delay deviations.

Based on the delay prediction error function and a constant α_{dv}, $0 < \alpha_{dv} < 1$, ($\alpha_{dv} = 0.25$ is recommended), the delay deviation can be estimated in Eq. 7.

$$\tilde{dv}_{\tau_k} = (1 - \alpha_{dv}) \cdot \tilde{dv}_{\tau_{k-1}} + \alpha_{dv} \cdot \left|ed_{\tau_k}\right| \quad (7)$$

QLDA relies exclusively on queue length to schedule packets. As such, it can easily be incorporated in packet scheduling schemes commonly used in current routers, such as FIFO, priority-based, and weighted fair queuing. Algorithm 1 describes the basic steps of the QLDA scheduling scheme.

IV. Energy Model

In this section, we consider a set of DVFS-enabled LCs and present a comprehensive energy model to determine the ***packet-based*** and ***router-based*** energy consumption, taking into consideration the frequency adjustment strategies used by the underlying scheduler.

A. Power Model

Two main components impact power consumption in network routers [6], [10], [11]. The first, referred to as static power, arises from the bias and leakage current to support control plan, environment units, and load-independent data plan. The second, referred to as dynamic power, results from the charging and discharging of the voltage saved in node capacitance of the circuit. We use P^S and P^D to denote static and dynamic power, respectively. In a router, NPUs operate in two possible states, namely "idle" and "busy". In the "idle" state, the power consumption is load-independent and equals to the static power, P^S. In the "busy" state, the power consumption is load-dependent and is composed of the static power P^S and dynamic power P^D. Consequently, the power consumed by a router can be expressed as follows:

$$P = \begin{cases} P^S, & \text{"idle" state} \\ P^S + P^D, & \text{"busy" state} \end{cases} \quad (8)$$

Algorithm 1 QLDA Scheduling Scheme.

For each NPU in LC at the router
Initialization:
$\overline{q}_{\tau_0}, \overline{d}_{\tau_0}, \tilde{dv}_{\tau_0} \leftarrow 0, k \leftarrow 1, f_{\tau_0} \leftarrow f_{Inital}$
Monitor queue length, q_{τ_k}, at the end time of τ_k
Update the queue-length smooth filter, $w^q_{\tau_k}$
Estimate the new average $\overline{q}_{\tau_k}$ for τ_k
$\overline{q}_{\tau_k} \leftarrow (1 - w^q_{\tau_k}) \cdot \overline{q}_{\tau_{k-1}} + w^q_{\tau_k} \cdot q_{\tau_k}$
Calculate d_{τ_k} based on $\overline{q}_{\tau_k}$
Estimate the new $\overline{d}_{\tau_k}$ and $\tilde{dv}_{\tau_k}$ for τ_k
Update the delay smooth filter, $w^d_{\tau_k}$
$\overline{d}_{\tau_k} \leftarrow (1 - w^d_{\tau_k}) \cdot \overline{d}_{\tau_{k-1}} + w^d_{\tau_k} \cdot d_{\tau_k}$
$ed_{\tau_k} \leftarrow d_{\tau_k} - \overline{d}_{\tau_k}$
$\tilde{dv}_{\tau_k} \leftarrow (1 - \alpha_{dv}) \cdot \tilde{dv}_{\tau_{k-1}} + \alpha_{dv} \cdot \left| ed_{\tau_k} \right|$
if $\overline{q}_{\tau_k} \leq q_l$ **then**
  Scale frequency f_{τ_k} to f_{min}
  $f_{\tau_k} \leftarrow f_{min}$
else if $\overline{q}_{\tau_k} \geq q_h$ **then**
  Scale frequency f_{τ_k} to f_{max}
  $f_{\tau_k} \leftarrow f_{max}$
else
  if $\left| \overline{d}_{\tau_k} - d^T \right| \leq \tilde{dv}_{\tau_k}$ **then**
    $f_{\tau_k} \leftarrow f_{\tau_{k-1}}$
  else
    $\overline{\lambda}_{\tau_k} \leftarrow A_{\tau_k} / \tau_k$
    $\rho_{\tau_k} \leftarrow \overline{\lambda}_{\tau_k} / u_{max}$
    Generate aggressivity factor, $\eta(\rho_{\tau_k})$
    Calculate scaling factor, ξ_{τ_k}
    $\xi_{\tau_k} \leftarrow (\frac{\overline{q}_{\tau_k} - q_l}{q_h - q_l})^{\eta(\rho_{\tau_k})}$
    Scale frequency, f_{τ_k}
    $f_{\tau_k} \leftarrow f_{min} + (f_{max} - f_{min}) \cdot \xi_{\tau_k}$
  end if
end if
$k \leftarrow k + 1$

Fixed Parameter:
- u_{max}: the maximal service rate at NPU.

Saved Variable:
- A_{τ_k}: the number of the arrival packets over τ_k.

The dynamic power, P^D, operated by the processor frequency, can be further expressed as $P^D = \gamma \cdot f^3$ [5], [12]. The parameter f denotes the clock frequency of the processor and γ is a constant parameter, expressed in units of $Watts/GHz^3$.

B. Packet-based Energy Model

For a given router, the dynamic power consumed by the data plane, P^D, is composed of two components, namely the *per-packet processing power* component, P_P, and the *per-byte store and forward power* component, $P_{S\&F}$ [11]. Both components are affected by the operational processor frequency, f. P_P represents the power consumed to process a given packet, regardless of the packet payload size. $P_{S\&F}$, on the other hand, represents the power needed to receive, store, switch and transmit a packet. Contrary to P_P, which only depends on the number of instructions needed to process a packet (IPP_P), $P_{S\&F}$ depends on the packet length, as packets with different lengths require different storage, switching time and transmission time, thereby consuming different amounts of power.

Let $IPB_{S\&F}$ denote the number of instructions required to process, store and forward a byte worth of data. Assuming a packet length of L $bytes$, the number of instructions required to process the packet is $IPP_{S\&F} = L \cdot IPB_{S\&F}$. Note that $IPB_{S\&F}$ is constant, as it only depends on the number of instructions to process a byte. Therefore, IPP_P can be expressed as a linear function of $IPB_{S\&F}$, namely $IPP_P = h \cdot IPB_{S\&F}$, where $h > 0$.

Let IPP represent the number of instructions to complete the processing, store, switch and transmission of an entire packet with length L by a NPU at a given LC. We have $IPP = IPP_P + IPP_{S\&F} = (h + L) \cdot IPB_{S\&F}$. The NPU's processing, storage, switching and transmission time of a packet, $T_p = \frac{IPP}{IPS}$, where IPS represents the number of instructions executed by the NPU per second. IPS can be further expressed as $\frac{f}{CPI}$, where f denotes the operational frequency of the NPU and CPI represents the number of cycles per instruction. Therefore, $T_p = \frac{IPP \cdot CPI}{f} = \frac{\Theta \cdot (h+L)}{f}$, where $\Theta = CPI \cdot IPB_{S\&F}$.

Let $f_{j,i}$ denote the operational frequency of the active NPU j in LC i. In the proposed scheduler, the derived frequencies may be continuous. In practice, however, the NPU only allows a number of manufacturer-specified discrete operational voltage levels, V=$\{V_1, ..., V_l, ..., V_M\}$. These discrete levels result in a corresponding set of discrete frequencies, F=$\{f_1, ..., f_l, ..., f_M\}$. Consequently, $f_{j,i}$ must be set to the smallest discrete frequency, $f_l (1 \leq l \leq M) | f_l \geq f_{j,i}$. The dynamic energy consumed by a successful packet transmission with length L at NPU j in LC i is given by:

$$E^D_{j,i}(T_p) = \gamma_{j,i} \cdot f^3_{j,i} \cdot T_p = \gamma_{j,i} \cdot \Theta_{j,i} \cdot f^2_{j,i} \cdot (h_{j,i} + L) \quad (9)$$

According to [11], the packet energy consumption can be expressed as $E_p = E_P + E_{S\&F} \cdot L$, where E_P, expressed in nJ/packet, denote the per-packet processing energy, and $E_{S\&F}$, expressed in nJ/byte, denote the per-byte store and forward energy [11]. Based on Eq. 9, we can compute $h_{j,i} = \frac{E_{P_{max_{j,i}}}}{E_{S\&F_{max_{j,i}}}}$ and $\gamma_{j,i} = \frac{E_{S\&F_{max_{j,i}}}}{\Theta_{j,i} \cdot f^2_{max_{j,i}}}$. Thus, the above packet-based dynamic energy consumption can be rewritten by Eq. 10.

$$E^D_{j,i}(T_p) = \frac{(E_{P_{max_{j,i}}} + E_{S\&F_{max_{j,i}}} \cdot L)}{f^2_{max_{j,i}}} \cdot f^2_{j,i} \quad (10)$$

C. Router-based Energy Model

Assume that a router is equipped with Ψ LCs, LC i, $1 \leq i \leq \Psi$, is equipped with n_i active NPUs. According to the power model, the energy consumption of the router, over T, can be expressed as $E(T) = E^S(T) + E^D(T)$, where $E^S(T)$ and $E^D(T)$ represent the energy consumed due to static power and dynamic power, repetitively, during time T. These energy components can be expressed as:

$$\begin{aligned} E^S(T) &= P^S \cdot T \\ E^D(T) &= \sum_{i=1}^{\Psi} \sum_{j=1}^{n_i} E^D_{j,i}(T) \end{aligned} \quad (11)$$

$E^D_{j,i}$ represents the energy consumed by NPU j in LC i due to dynamic power, over T. Let $Z_{j,i}$ be the amount of time

interval τ at NPU j in LC i over T, and $\tau_1, \cdots, \tau_k, \cdots, \tau_{Z_{j,i}}$, $1 \leq k \leq Z_{j,i}$, represent the frequency time slots at NPU j in LC i. Assuming f_{j,i,τ_0} is the initial frequency. The frequency of NPU j at LC i, over the k^{th} time slot, τ_k, is $f_{j,i,\tau_{k-1}}$, where $1 \leq i \leq \Psi$, $1 \leq j \leq n_i$ and $1 \leq k \leq Z_{j,i}$. Let D_{j,i,τ_k} denote the number of packets serviced by NPU j in LC i over the interval τ_k, and $\overline{L}_{j,i}$ represent the average length of the packets serviced at NPU j in LC i. According to Eq. 10, the total dynamic energy consumed by NPU j in LC i over T can be expressed as:

$$E^D_{j,i}(T) = \sum_{k=1}^{Z_{j,i}} \frac{(E_{P_{max_{j,i}}} + E_{S\&F_{max_{j,i}}} \cdot \overline{L}_{j,i}) \cdot D_{j,i,\tau_k}}{f^2_{max_{j,i}}} \cdot f^2_{j,i,\tau_{k-1}} \tag{12}$$

Therefore, the entire energy consumption of the router over T can be derived as:

$$E(T) = P^S \cdot T + \sum_{i=1}^{\Psi} \sum_{j=1}^{n_i} \sum_{k=1}^{Z_{j,i}} \frac{(E_{P_{max_{j,i}}} + E_{S\&F_{max_{j,i}}} \cdot \overline{L}_{j,i}) \cdot D_{j,i,\tau_k}}{f^2_{max_{j,i}}} \cdot f^2_{j,i,\tau_{k-1}} \tag{13}$$

Eq. 13 demonstrates that adjusting the frequency, as opposed to using the maximum frequency, further reduces the energy consumption. The following simulation study will be used to further explore the impact of dynamically adjusting frequencies on the energy consumption.

V. Simulation and Analysis Framework

In this section, we present a simulation framework to assess the performance of the Delay-aware scheduling scheme discussed. The energy model is incorporated in a NS2-based network simulation platform to carry out an extensive performance analysis study of the proposed scheduler, focusing on: (i) sensitivity of the proposed scheme to critical design parameters; (ii) comparative analysis of the proposed scheduler's performances to other similar schedulers, in different network environments and traffic loads.

A. Simulation Environment

In our simulation framework, we consider two network topology models: i.e. ***dumbbell*** and ***parking lot***, as displayed in Fig.3, which are two promising models to capture the behavior and performance of a large variety of Internet applications [13]–[16]. S and D denote end-hosts, and the intermediate nodes between S and D are energy-saving routers. The capacities of links between all the routers are $10\ Gbps$. The routers implement FIFO scheduling and DropTail queuing. The propagation delays between the sources and the destinations are $30\ ms$. In each router, all LCs are configured with multiple NPUs, each using a specific QoS-aware DVFS scheduler. In order to simulate real scenarios, Huawei CX600-X3 Metro Router model [11], supporting $10GE$ LCs, is used. We further assume that each $10GE$ port provides $250\ ms$ worth of traffic buffering. This results in processor buffers of approximately $250ms \times 10Gbps$, which is roughly 250000 packets, assuming the average packet size of 1250 bytes. The range of operating frequencies, $[1.6GHz,\ 2.4GHz]$, for a given NPU, is based on Intel XEON DPDK specification [17].

TABLE I: Main simulation parameters and conditions.

Items	Simulation Parameters	Simulation Conditions
Router	Router Node	Metro Router [11]
	NIC port	$10GE$
	Operating Frequency (GHz)	$1.6 \sim 2.4$
	$CPI(cycles/instruction)$	1.2
	$EP_{P_{max}}(nJ/pkt)$	1375 [11]
	$EB_{S\&F_{max}}(nJ/byte)$	14.4 [11]
	$P^S(Watts)$	352 [11]
Packet	Packet Max size $(bytes)$	1500
	$IPB(instructions/byte)$	1.5
Queue	Service Discipline	FIFO
	Queuing Discipline	DropTail
Network	Topology Model	3-hop dumbbell
		4-hop parking lot
	Network Traffic Load ρ	$0.1, \cdots, 0.95$
	Propagation Delay (ms)	dumbbell: 30; parking lot: 30
	Traffic Model	Video:VoIP:Exponential

TABLE II: Traffic source models and specifications.

Flow Type	Load Percentage	T_{On} (ms)	T_{Off} (ms)	Peak Rate	β
Video	50%	NA	NA	$10Mbps$	NA
VoIP	20%	400	400	$64Kbps$	1.1
Exp VBR	30%	40	360	$256Kbps$	NA

TABLE III: Four combinations of (q_l, q_h).

	$q_h = 60\% \cdot Q$	$q_h = 80\% \cdot Q$
$q_l = 4\% \cdot Q$	(1.0 E+4, 1.5 E+5)	(1.0 E+4, 2.0 E+5)
$q_l = 10\% \cdot Q$	(2.5 E+4, 1.5 E+5)	(2.5 E+4, 2.0 E+5)

TABLE I describes the main simulation parameters used in this simulation study. According to [18], [19], TABLE II specifies the traffic source models, namely one constant bit rate (CBR) and two variable bit rate (VBR) models, including Pareto or Exponential On/Off distribution traffic. TABLE IV lists four combinations of q_l and q_h. In addition to the proposed QLDA scheduler and the Load-aware scheduler, EWMAP [9], we also implemented a generic scheduler, referred to as NoDVFS, that operates network devices at their maximum frequency. NoDVFS provides a baseline to compare the performance of energy-aware schedulers. Two QoS-aware DVFS-based schemes, i.e. QLDA and EWMAP, are compared in this paper.

The ITU G.114 specification recommends less than $150\ ms$ one-way end-to-end delay for high-quality real-time traffic such as voice and video. Therefore, we consider the following QoS requirements [20], to evaluate energy saving percentage (ESP), average end-to-end delay (AED) of the all schemes in our simulation:

- $150\ ms$ as the one-way average end-to-end delay threshold,
- $10\ ms$ as the delay jitter bound (DJB),
- 1% as as the packet loss rate (PLR) threshold.

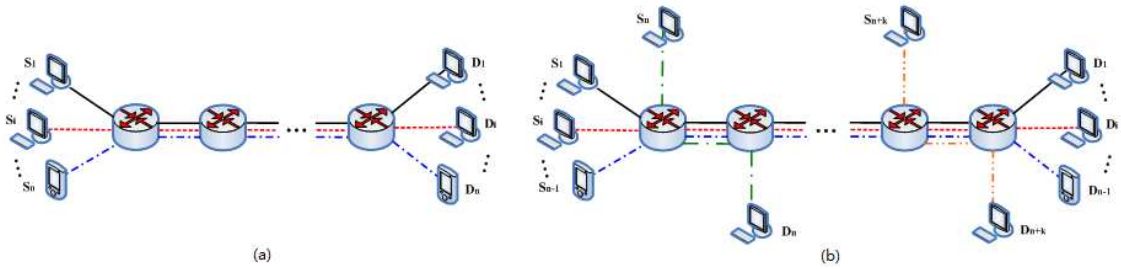

Fig. 3: Two network topology models: (a) dumbbell, and (b) parking lot.

TABLE IV: Impact of η on ESP, AED, DJB and PDMRT of QLDA scheme with $q_l : q_h = 4\% : 80\%$.

	Dumbbell model			**Parking lot model**		
Load ρ	0.7	0.8	0.9	0.7	0.8	0.9
$ESP(\%)$	9.82	7.83	4.76	9.76	8.68	6.16
$AED(ms)$	122.30	133.89	132.34	116.89	131.18	125.77
$DJB(ms)$	1.73	3.40	6.12	1.24	2.78	5.74
$PLR(\%)$	0	0	0	0	0	0

B. Sensitivity to the main parameters of QLDA

In this section, we carried out a series of experiments to do the sensitivity analysis of the proposed QLDA scheme to different parameters.

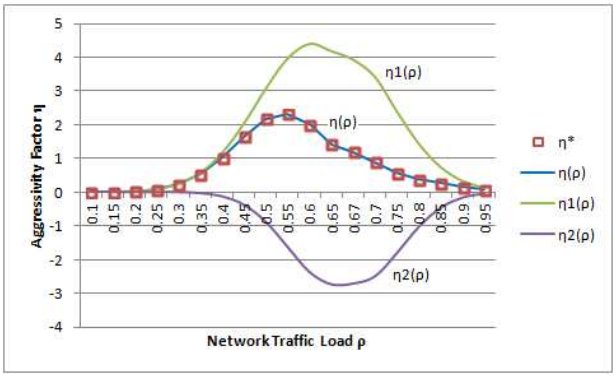

Fig. 4: The aggressivity factor $\eta(\rho)$.

1) Sensitivity to η: The first experiment is designed to study the sensitivity of the scheduler to the aggressivity factor η. The results show that the value of the aggressivity factor to achieve the highest energy saving depends on the network load. In order to determine the "optimum" η^*, a series of simulation experiments where carried out, whereby for a given network load, ρ, multiple values of η are tested and the value which produces the highest energy saving, without violating the traffic QoS requirements, is selected. The experiments used two different network models, namely dumbbell and parking lot, assuming $f_{min} = 1.6\ GHz$ and $f_{max} = 2.4\ GHz$. Using Matlab, the two independent variables, η and ρ, are fitted by a Gaussian process regression (GPR) function of successive approximations, as illustrated Eq. 14. In this equation, (a_1, b_1, c_1) and (a_2, b_2, c_2) are GPR model parameters, where $(a_1, b_1, c_1) = (4.4, 0.6085, 0.1805)$ and $(a_2, b_2, c_2) = (-2.745, 0.6554, 0.1466)$. The fitted curve is depicted in Fig.4.

$$\begin{aligned}\eta(\rho) =& \eta_1(\rho) + \eta_2(\rho) \\ =& a_1 \cdot e^{-(\frac{\rho - b_1}{c_1})^2} + a_2 \cdot e^{-(\frac{\rho - b_2}{c_2})^2}\end{aligned} \tag{14}$$

Using the load-dependent values of η, generated by the GPR function, the two network topology models, dumbbell and parking lot, were used to assess the performance of QLDA and determine the levels of energy saving it achieves, under different network loads, while maintaining acceptable QoS requirements. The results of these experiments are shown in TABLE IV.

2) Sensitivity to τ: The second experiment is designed to study the scheduler's sensitivity to the rate of DVFS adjusting. The values of η for different traffic loads refer to TABLE IV. Under a frequency range, $[f_{min}, f_{max}]$, a small frequency adjustment interval creates more opportunities for a more accurate adjustment of the frequency, based on the queue length. A small interval, however, increases the frequency adjustment overhead. A large frequency adjustment interval reduces the overhead required to adjust frequencies, but fails to capture more accurately the current level of congestion. Fig.5 (a) and (b) depict the energy-saving percentage and the corresponding average end-to-end packet delay for the different network models, using different frequency adjustment interval, τ, under the range of $[0.01, 100]\ \ ms$. The results show that QLDA, assuming $q_l = 4\%? \times Q$ and $q_h 80\% \times Q$, is not sensitive to τ when the value of τ is under $1\ ms$. Therefore, $\tau = 1\ ms$ is selected for the rest of experiments.

3) Sensitivity to c_q: QLDA scheme uses EWMA based algorithm with weight, w^q, to predict the queue length. The constant parameter c_q is used in the error prediction function to adaptively adjust w^q. Different values of c_q in the range $[0.01, 0.50]$ are tested in the QLDA scheme. The results show that the energy saving and the average end-to-end packet delay are not sensitive to c_q. When the value of c_q increases,

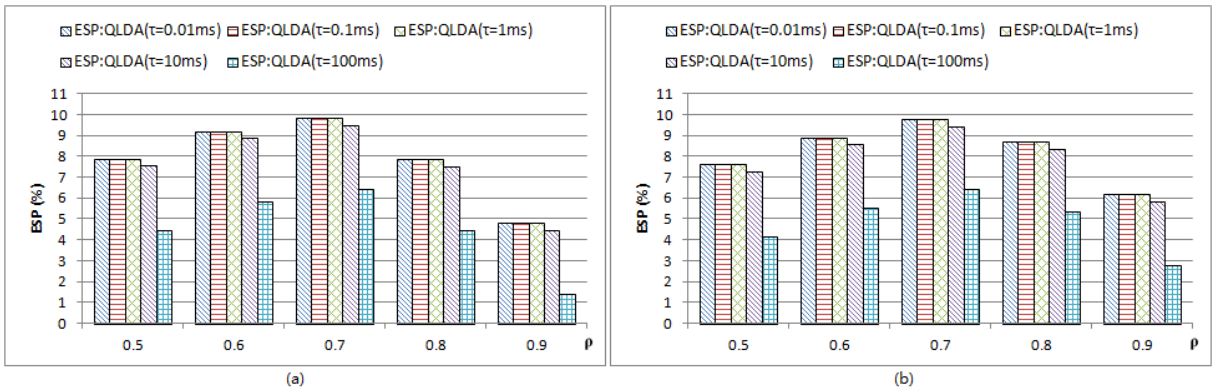

Fig. 5: ESP comparisons for QLDA with different τ in (a) dumbbell model, and (b) parking lot model.

TABLE V: Impact of $q_l : q_h$ on ESP and AED in the QLDA scheme under dumbbell model.

	ESP (%)				AED (ms)			
$q_l : q_h$	$4\% : 60\%$	$4\% : 80\%$	$10\% : 60\%$	$10\% : 80\%$	$4\% : 60\%$	$4\% : 80\%$	$10\% : 60\%$	$10\% : 80\%$
$\rho = 0.7$	9.75	9.82	9.77	9.85	111.19	122.23	143.33	150.77
$\rho = 0.8$	7.73	7.83	7.75	7.85	117.90	133.89	162.18	177.65
$\rho = 0.9$	4.60	4.76	4.63	4.76	117.03	132.34	158.62	174.17

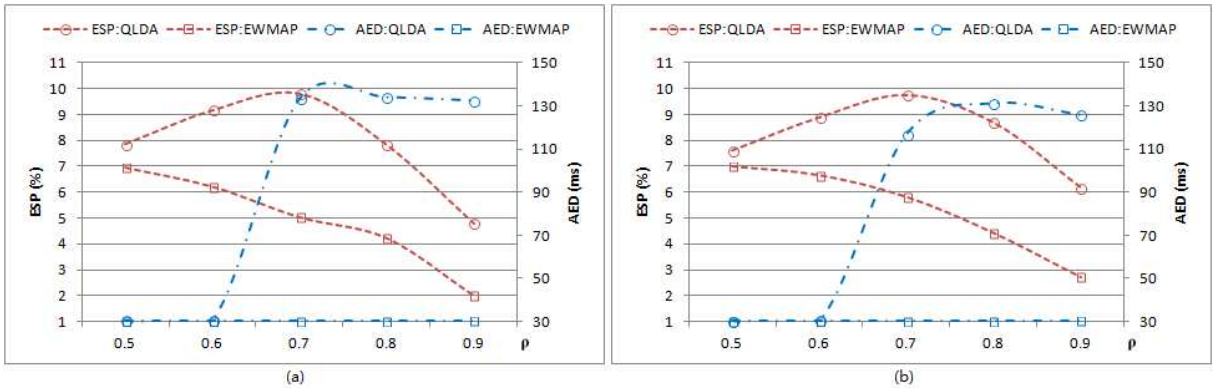

Fig. 6: ESP and AED comparisons between QLDA ($q_l : q_h = 4\% : 80\%$) and EWMAP ($\mu = 0.2$) for (a) dumbbell model, (b) parking lot model.

the energy saving increases slightly. Therefore, $c_q = 0.5$ is selected for the following analysis.

4) Sensitivity to c_d: Similarly, QLDA uses the EWMA algorithm to predict the average packet delay. The constant parameter c_d is used in the error prediction function to dynamically adjust the smooth filter w^d. Different values of c_d in the range $[0.01, 0.5]$ are tested in the QLDA scheme. The results show that the energy saving and the average end-to-end packet delay are not sensitive to c_d. When the value of c_d decreases, the energy saving increases slightly. Therefore, $c_d = 0.01$ is considered.

5) Sensitivity to q_l and q_h: In this experiment, the value setting of η for different traffic load is based on the above GPR function, as shown in Fig.4, and the value of τ is set to $1\ ms$, while the thresholds, q_l and q_h, are varied, as described in TABLE III. Four combinations of q_l and q_h are tested to study the impact of the queue length thresholds on energy saving and average packet delay. The results, shown in TABLE V, indicate that, although the energy saving is not highly sensitive to q_h, a higher value of q_h leads to higher energy saving. The results also show that packet delay is highly sensitive to q_l, as the delay increases dramatically when the value of q_l increases. Therefore, adjusting queue-length thresholds can improve the effectiveness and efficiency of the QLDA scheme. The results show that for a dumbbell topology, the ratio $q_l : q_h = 4\% : 80\%$ leads to the highest energy savings, without violating QoS requirements. A similar outcome can be observed in the case of a parking lot model.

C. Comparative analysis

In [9], Mandviwalla et al. propose three Load-aware predictors to reduce energy consumption in LCs, in which the most effective Load-aware predictor, called EWMAP, uses EWMA algorithm with a fixed load smooth filter, μ ($\mu = 0.2$ is recommended in the EWMAP scheme [9]), to predict traffic

load over a constant perdition interval, τ (i.e. PI in [9]), to control the execution rates of LCs, aiming to achieve energy saving. In the Load-aware schemes, choosing a small prediction period in the Load-aware schemes could suffer the overhead impact of the back-to-back undesirable DVFS adjustment. Different values of the prediction period, τ, in the same range $[0.01, 100]\ ms$ as the QLDA scheme are tested to determine sensitivities of the EWMAP scheduler to DVFS adjustment. Different from QLDA, the EWMAP scheme exhibits sensitivity to τ. The results show that the EWMAP scheme can achieve the largest energy saving without QoS violence when τ is set as $1\ ms$.

Using the same router-based energy model, we compare the proposed QLDA scheme with the EWMAP scheme under $\tau = 1\ ms$ in two different network models, as shown in Fig.6 (a) and (b). We found that two network models have same trend in the energy saving and the average end-to-end packet delay in both QoS-aware schemes respectively. The results show that these two QoS-aware schemes are potential to save significant energy. Under the same QoS requirements, the QLDA scheme with $q_l : q_h = 4\% : 80\%$ can provide up to 9.82% energy saving with AED of $122.30\ ms$ and DJB of $1.73\ ms$, and 9.76% energy saving with AED of $116.89\ ms$ and DJB of $1.24\ ms$, in the dumbbell model and the parking lot model, respectively. Although the EWMAP scheme leads to an increase in energy saving, from 2% to 7%, under different traffic loads, the results show that QLDA achieves up to 5% increase in energy saving than EWMAP, without violating QoS requirements.

VI. Conclusion

In this paper, we propose a Delay-aware DVFS-based scheduler, which scales frequency and achieves energy saving, while meeting QoS requirements. When and how decisions are made to adjust the router execution rates are based on the predicted queue-length and the target packet delay. A simulation framework, including a comprehensive router energy model, which accounts for both static and dynamic energy consumption, is proposed to investigate and compare the performance of the proposed scheme to other similar QoS-aware DVFS-based schemes. Different networking topologies and traffic models are used to carry out sensitivity analysis of the proposed scheme with respect to its main parameters, assess its performance in different network environments, and perform comparative analysis with other schemes. The simulation results show that the proposed QLDA scheme achieves high energy-saving, without violating the QoS requirements of the supported applications. More specifically, the results show that QLDA outperforms load-aware energy-saving schemes and can achieve up to 10% energy saving, while meeting the desired QoS performance. Overall, the results indicate that load-aware schemes, such as EWMAP, are limited when it comes to achieving energy savings without violating QoS requirements. On the other hand, queue length based, delay-aware schedulers, such as QLDA, using adequate queue length thresholds and load-dependent aggressivity factors, can control frequency scaling to achieve a balance between energy saving and QoS requirements.

Acknowledgment

This material is based in part upon work supported by the National Science Foundation under Grants Number CNS-011418 and CNS-1162159. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 47, 2014.

[2] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008*, pp. 1130–1138.

[3] L. J. Wobker. (2012) Power consumption in high-end routing systems. [Online]. Available: https://www.nanog.org/meetings/nanog54/presentations/Wednesday/Wobker.pdf

[4] H. Imaizumi and H. Morikawa, "Directions towards future green internet," in *Towards Green Ict*. Niels Jernes Vej 10, 9220 Aalborg, Denmark: River Publishers, 2010, vol. 9, pp. 37–53.

[5] M. E. T. Gerards, "Algorithmic power management: Energy minimisation under real-time constraints," Ph.D. dissertation, Centre for Telematics and Information Technology, University of Twente, 2014.

[6] G. L. Valentini, W. Lassonde, S. U. Khan, N. Min-Allah, S. A. Madani, J. Li, L. Zhang, L. Wang, N. Ghani, J. Kolodziej *et al.*, "An overview of energy efficiency techniques in cluster computing systems," *Cluster Computing*, vol. 16, no. 1, pp. 3–15, 2013.

[7] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation." in *NSDI*, vol. 8, 2008, pp. 323–336.

[8] R. Tucker, J. Baliga, R. Ayre, K. Hinton, and W. Sorin, "Energy consumption in ip networks," in *ECOC Sym. on Green ICT*, 2008, p. 1.

[9] M. Mandviwalla and N.-F. Tzeng, "Energy-efficient scheme for multiprocessor-based router linecards," in *Applications and the Internet, 2006. SAINT 2006. International Symposium on*. IEEE, 2006, pp. 8–pp.

[10] J.-M. Pierson, *Large-scale Distributed Systems and Energy Efficiency: A Holistic View*. 111 River Street, Permissions Department, Hoboken, NJ 07030, USA: John Wiley & Sons, 2015.

[11] A. Vishwanath Member, K. Hinton, R. Ayre, and R. Tucker, "Modeling energy consumption in high-capacity routers and switches," *IEEE JSAC*, vol. 32, no. 8, pp. 1524–1532, 2014.

[12] X. Chen, X. Liu, S. Wang, and X.-W. Chang, "Tailcon: Power-minimizing tail percentile control of response time in server clusters." in *SRDS*, 2012, pp. 61–70.

[13] D. Hayes, D. Ros, L. Andrew, and S. Floyd. (2014) Common tcp evaluation suite. [Online]. Available: https://tools.ietf.org/html/draft-irtf-iccrg-tcpeval-01

[14] E. Jonckheere, K. Shah, and S. Bohacek, "Dynamic modeling of internet traffic for intrusion detection," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3. IEEE, 2002, pp. 2436–2442.

[15] K. Shah, S. Bohacek, and E. A. Jonckheere, "On the predictability of data network traffic," in *Proceedings of the American Control Conference*, vol. 2, 2003, pp. 1619–1624.

[16] Z. Móczár, S. Molnar, and B. Sonkoly, "Multi-platform performance evaluation of digital fountain based transport," in *SAI, 2014*. IEEE, 2014, pp. 690–697.

[17] Intel. (2012) Data plane development kit overview. [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/presentation/dpdk-packet-processing-ia-overview-presentation.pdf

[18] W. Simpson, *Video over IP: a practical guide to technology and applications*. 711 3rd Avenue, New York, NY 10017, USA: Taylor & Francis, 2006.

[19] [Online]. Available: http://www.voip-info.org/wiki/view/Codecs

[20] T. Szigeti and C. Hattingh, *End-to-end qos network design*. 800 East 96th Street, Indianapolis, Indiana 46240, USA: Cisco press, 2005.