

An Energy-aware Routing Mechanism for Latency-sensitive Traffics

Xiao Peng^{1*}, Qu Peixin² and Qu Xilong¹

¹*Department of Computer and Communication, Hunan Institute of Engineering*

²*School of Information & Engineering, Henan Institute of Science and Technology*

*xtanefn@gmail.com

Abstract

With the rapid development of Internet technology and enhanced QoS requirements, network energy consumption has attracted more and more attentions due to the overprovision of network resources. Generally, energy saving can be achieved by sacrificed some performance. However, many popular applications require real-time or soft real-time QoS performance for attracting potential users, and existing technologies can hardly obtain satisfying tradeoffs between energy consumption and performance. In this paper, a novel energy-aware routing mechanism is presented with aiming at reducing the network energy consumption and maintaining satisfying QoS performance for these latency-sensitive applications. The proposed routing mechanism applies stochastic service model to calculate the latency-guarantee for any given network links. Based on such a quantitative latency-guarantee, we further propose a technique to decide whether a link should be powered down and how long it should be kept in power saving mode. Extensive experiments are conducted to evaluate the effectiveness of the proposed mechanism, and the results indicate that it can provide better QoS performance for those latency-sensitive traffics with improved energy-efficiency.

Keywords: Energy Efficiency; Network Traffic; Optional Routing; Stochastic Model

1. Introduction

Recently, high energy consumption of Internet has become a critical concern with the rapidly increasing of various kinds of high-end applications, which often impose more-stringent QoS requirements onto the backbone networks [1, 2]. As a result, more and more network devices have been equipped into various kinds of networks even most of them seldom work in full utilization state. Unlike those end-user IT devices (*i.e.*, CPU, disk, display and *etc.*) that can be configured into various energy saving states when there is no task available, networking equipments typically should be set in full power modes with aiming to maximize throughput as well as minimize latency, which further decrease the energy-efficiency of the Internet. As noted in [3, 4, 5], the average link utilization is less than 30-40% and the duration of peak traffic load is only a small fraction of the entire day.

To deal with the high energy consumption problem in the Internet, plenty of efforts have been taken into energy conserving from different aspects. The proposed approaches can be briefly categorized into three levels: (1) At individual switch or router level, some of the subcomponents such as line-card can be configured as sleeping mode when they are idle [6, 7]; (2) At the network level, multiple traffics can be aggregated dynamically at runtime by some protocols, which are often designed for

high-performance distributed computing environments [8, 9, 10]; (3) At the topology level, adaptive and dynamical routing mechanism can be applied to satisfying the requirements of network traffics [11, 12]. Unfortunately, most of the above approaches work in coarse-grained time scales, that is, they can not quickly take response according to the changing of traffic workloads in terms of energy consumption.

In this work, we take our efforts on designing a novel routing mechanism for delivering latency-sensitive traffics, because most of nowadays applications require real-time or soft real-time QoS performance for attracting potential users [13, 14]. It is a well-known fact that better QoS performance will inevitably lead to more energy consumption. Therefore, some tradeoffs between QoS performance and energy-efficiency should be taken. Unlike many existing work that using heuristic algorithm for routing [15, 16, 17, 18], we apply stochastic service model to evaluate the deadline-guarantee for any given route. By such a quantitative deadline-guarantee, we propose a technique to decide whether a link should be powered down and how long it should be kept in power saving mode.

The rest of this paper is organized as follows. Section 2 presents the related work. In Section 3, we present some definitions and the problem description; In Section 4, we present our deadline-guarantee calculating technique and the energy-aware routing mechanism. In Section 5, experiments are conducted to investigate the effectiveness of the proposed mechanism. Finally, Section 6 concludes the paper with a brief discussion of the future work.

2. Related Work

At hardware level, to meet stringent throughput and latency requirements, routers need to employ multiple off-chip memory components in hierarchical configurations [15]. For example, Micron [19] consists of a few Gigabytes DRAM banks and some power calculators to measuring its energy consumption. To provide more accurate power measurements, some packet-based power models have been designed to monitor router's online power consumption at various workload levels [6, 7, 8]. For instance, Cisco's CRS-1 platform has reported that of the power consumption of a line-card is 375W, DRAM consumes 72W, and buffer memory consumes about 10% of the total power consumption [20]. Generally speaking, the hardware energy saving approaches can be classified as: sleeping mode and rate controlling. The works in [21, 22] belong to rate controlling technique, which enable the links adjusting the rate to satisfying the real-time traffic. In [9, 21], opportunistic sleeping technique is evaluated in two practical network environments. In [9, 22], several heuristics are proposed by using sleeping mode switching.

At the IP level, many solutions have been proposed for energy conserving. For instance, in [23], the authors proposed a heuristic to compute the maximum number of IP links and routers to be powered off for a given traffic matrix by estimating the power consumption of nodes and links; In [24], a traffic engineering solution is presented with aiming to minimize the energy consumption among a set of paths; In [25, 26], the authors designed two techniques to minimize the energy consumption of the links in overlaying networks. The above approaches generally resort to certain optimization programming technique to find the solutions. Therefore, the complexity of these approaches might be very high due to feasibility tests.

At the protocol level, extensive protocols have taken energy-consumption into account. For example, Power Aware Multi-Access protocol [27] is an adaptation of

IEEE 802.11, in which the wakeup radio parameter is significantly lower than that of the radio transmission so as to reduce the aware period. S-MAC [28] is a protocol developed to address the energy issue in the sensor networks, building on contention based protocols like IEEE 802.11. It follows a simple scheduling scheme that allows neighbors to sleep for long periods and to synchronize wakeups. T-MAC [29] extends S-MAC by adjusting the length of time sensors are awake between sleep intervals based on communication of neighbors. In this way, less energy is wasted due to idle listening when traffic is light. It is clear that all the protocols are for wireless networks instead of Internet, since the energy concern in wireless network emerges much earlier than that in Internet.

3. Problem Description

In current networks, most of the network links have three power modes, including ACTIVE, SLEEP, and D-SLEEP. In ACTIVE mode, the real-time traffic is near to the upper bound of the bandwidth which makes all the equipments have to work in full power state. In SLEEP mode, the traffic is near to the low bound of the bandwidth, therefore some equipment (*i.e.*, routers and switches) can be turn into power saving modes so as to obtain energy conservation. Generally speaking, the latency of the link in SLEEP mode will be slightly increased comparing and switching time between ACTIVE and SLEEP is about 0.01 ms to 0.05 ms [30]. When a link is in D-SLEEP mode, it can be considered as working in idle mode and most of the equipments can be turn in power saving even power off modes. In such a mode, the power consumption is lowest and the network latency is highest in all cases. Normally, the switching time between D-SLEEP and SLEEP is about 1 ms to 5 ms.

In existing works, most studies take efforts on how and when changes between those three modes according to the tradeoff between performance and energy-efficiency. For example, a simple way of energy saving is switching a link from ACTIVE to SLEEP/D-SLEEP when its utilization is below a predefined value. However, this sleeping scheme causes high latencies due to frequent changes is often required especially when the traffic is very dynamically. In addition, such an approach ignores the different QoS requirements in the traffic because all the packets in the traffic will be affected when the power mode of a link is changed. To quantitatively evaluate the intensiveness of network workload as well as their individual QoS requirements, we applied stochastic model [31] to describe the working of the network links. Here, we present some definitions that will be used in the following sections.

Definition 1. A link is noted as L_i which is characterized as a three-tuple $\langle \lambda_i, \mu_i, c_i \rangle < \lambda_i, \mu_i, c_i \rangle$, where λ_i represents the average arrival interval between two packets, μ_i represents the average transferring time, and c_i is the maximal packet that L_i can concurrently service.

Definition 2. A packet p_i is characterized a two-tuple $\langle L, d \rangle$, where $L = \{L_1, \dots, L_n\}$ is the link table of the packet from source to the destination, and d is the total latency specified by the packet.

Definition 3. Given a link and a period of time, its bandwidth allocation scheme is defined as a mapping that noted as $M: B(t) \times p(t) \rightarrow \{0,1\}$, where $B(t)$ is the available bandwidth at time t and $p(t)$ is packet arrives at time t .

Definition 4. The network is noted as a weighted graph $G=<V, E>$, where V is set of routers and E is the set of all links.

By the above definitions, the goal of our approach is to minimize the energy consumption of the network equipments while maximizing the achieved end-to-end network performance for latency-sensitive traffic. By the abovementioned power modes, we can see that the traffic load is the key factor that we should adjust for achieving the goal. However, the most challenging issue is that whether a link can satisfy the QoS requirement of those latency-sensitive packets if we put the link is SLEEP or D-SLEEP mode. In some previous studies, traffic prediction is a possible approach to obtaining some optimization results. However, such a prediction often results in high error because of dynamically traffics in current networks. In this work, our approach does not rely on any traffic prediction techniques. On the contrary, we present a technique that can evaluate the probability that a given link can satisfy the latency requirement of any packet. Based on this technique, we can monitor all the links and decide that which of them should be powered up and which of them should be powered down.

4. Energy-aware Routing Mechanism

4.1. Calculation of Latency Guarantee

According to the Definition 1, we can use M/M/C model to describe the working of a link. Therefore, the arrival of packets at link L_i is a Possion process with mean value λ_i , and the service time follows exponential distribution with a mean value μ_i , and workload intensive of L_i can be noted as $\rho_i=\lambda_i/(\mu_i c_i)$. By this M/M/C, we first need to evaluate that whether a link can satisfy the QoS requirement of the packets that arrives. Since we mainly focus on latency-intensive packet, the QoS metric we consider is the latency requirement d as noted in Definition 2.

Theorem 1. If a link L_i is modeled as M/M/ c_i stochastic service model, the probability that the packet arrive at L_i can be transferred within d_i is as following

$$\Pr\{\omega \leq d_i\} = \sum_{n=0}^{c_i} \delta \cdot \frac{(\rho_i \cdot c_i)^n}{n!} + \sum_{k=1}^{c_i \cdot \mu_i \cdot d_i - 1} \delta \cdot \frac{\rho_i^{k+c_i} \cdot c_i^{c_i}}{c_i!}$$

where $\delta = \left[\sum_{n=1}^{c_i} \frac{(\rho_i \cdot c_i)^n}{n!} + \frac{(\rho_i \cdot c_i)^{c_i}}{c_i} \frac{1}{1-\rho_i} \right]^{-1}$, ω is the random variable representing the actual latency of the packet, d_i is the latency on link L_i .

Proof. Let ψ be a random variable representing the number of waiting packet in L_i . According to stochastic model theory [31], the probability that there are k waiting packets in L_i is

$$\Pr\{\psi=k\} = \begin{cases} \delta \cdot \frac{\rho_i^{k+c_i} \cdot c_i^{c_i}}{c_i!}, & k > 0 \\ \sum_{n=0}^{c_i} \delta \cdot \frac{(\rho_i \cdot c_i)^n}{n!}, & k = 0 \end{cases} \quad (1)$$

$$\text{where } \delta = \left[\sum_{n=1}^{c_i} \frac{(\rho_i \cdot c_i)^n}{n!} + \frac{(\rho_i \cdot c_i)^{c_i}}{c_i} \frac{1}{1-\rho_i} \right]^{-1} \quad (2)$$

By $M/M/c_i$ model, the service rate is $\mu_i c_i$, which means the amount of packets that L_i can complete in a unit time is $c_i \mu_i$. So, the amount of packets that L_i can complete in period d_i is $\mu_i c_i d_i$. Therefore, the probability that L_i can guarantee the latency requirement of this packet is equal to the probability that the number of waiting packet in L_i is not more than $\mu_i c_i d_i - 1$. That is

$$\Pr\{\omega \leq d_i\} = \Pr\{\psi \leq c_i \mu_i d_i - 1\} \quad (3)$$

By (1)(2)(3), we can get that

$$\begin{aligned} \Pr\{\omega \leq d_i\} &= \Pr\{\psi \leq c_i \mu_i d_i - 1\} \\ &= \sum_{k=0}^{c_i \cdot \mu_i \cdot d_i - 1} \Pr\{\psi = k\} \\ &= \sum_{n=0}^{c_i} \delta \cdot \frac{(\rho_i \cdot c_i)^n}{n!} + \sum_{k=1}^{c_i \cdot \mu_i \cdot d_i - 1} \delta \cdot \frac{\rho_i^{k+c_i} \cdot c_i^{c_i}}{c_i!} \end{aligned} \quad (4)$$

■

By the conclusion of Theorem 1, we can easily figure out that how much a link can guarantee the latency requirements of the packets. In common sense, we can set a throttle below which we will set the network equipments that connecting with this link into power active modes. Further mode, we can also define multiple throttles that can be used to control the actual power modes of network equipments. However, such a coarse-grained mechanism might result in unpredictable network performance, especially for those backbone networks which are expected to provide stable QoS service for various kinds of applications.

So, we propose a novel routing mechanism, namely Energy-aware Routing Mechanism (ERM), to overcome the above the problem. The main idea can be summarized as: (1) the routers will monitor the latency guarantees of the links in real-time manner by using the conclusion of Theorem 1; (2) As soon as they notice that some links can only provide low latency guarantees for the arrival packets, they will modify their routing tables and try to avoid using those links as the next routing items for a given period of time; (3) when a link continues to service with high latency guarantee, its connected network equipments will be notified for changing their power modes if possible. In the next section, we will present the detail implementation of this mechanism.

4.2. Energy-aware Routing Mechanism

The goal of our ERM is to minimize the energy consumption of the access router while maximizing the achieved end-to-end network performance for latency-sensitive packets. To do this, firstly we should provide an effective approach to monitoring the links' latency guarantee. As the routers are sensitive to any extra performance costs, it is impossible to calculate the $\Pr\{\omega \leq d_i\}$ value for all packets. In fact, we can achieving the same objective by monitor the queue length variations of the output buffer and, based on them, determine the service rate variation that should be applied to keep the

queue length as constant as possible. For example, given a latency guarantee value we can figure out its waiting queue length.

Due to the natural burstiness of traffic sending rate, the variation registered in the output buffer queue length between two generic time instants should be averaged over duration equal to the round-trip time. So, the first issue is how to estimate the round-trip time of traffic flows crossing the router. In addition, the well-known saw-tooth behavior of the TCP sending rate causes oscillations in the buffer queue length. This can trigger frequent service rate transitions that can lead to instability, resulting in performance degradations. For the above reasons, we propose to the following rules to stabilize the output buffer queue as possible as we can:

- ◆ If the output buffer input rate is higher than the service rate, we applies C-EARTH [32] algorithm to determines the desired output capacity by monitoring the output queue length variations and calculating the service rate variation that should be applied to keep the queue length as constant as possible and make the service rate equal to the arrival rate.
- ◆ If the output buffer input rate is less than the service rate, we use AWM [33] to increase the traffic sending rate.

As to the modification of routing tables, we design an optional forwarding routing mechanism which periodically tries to increase the usage of active links and to increase the down time of already sleeping links. Optional forwarding mechanism uses only local information because it should combine with hardware scheme operating at fine-grained time granularities. For example, to links with a static single-path routing, the exit latency is caused whenever the primary next-hop is in sleeping. However, if the optional next-hop is in ACTIVE or SLEEP, a packet could be forwarded to the optional next-hop, instead of forwarding to the primary next-hop in D-SLEEP. In summary, optional forwarding is to keep using ACTIVE and SLEEP links as much as possible, instead of those in D-SLEEP modes. When using optional forwarding routing mechanism, it might cause routing loops as well as increased routing path. To solve this problem, we implement the optional forwarding routing that is allowed to select ECMP paths [34]. Therefore, it can limit the increment of path length and exit latencies.

4.3. Router Energy Optimization

As noted in Section 4.1, the network equipments should switch their power mode when some links are working in different modes (ACTIVE, SLEEP or D-SLEEP). In this section, we mainly concentrate on the router devices since it contribute the major part of energy consumption among all network equipments. It is well known that current routers can be set into different power modes by administrator. In hardware level, such a mechanism is implemented by dynamically changing the size and number of packet buffers. In Figure 1, we present a popular buffer architecture of nowadays routers.

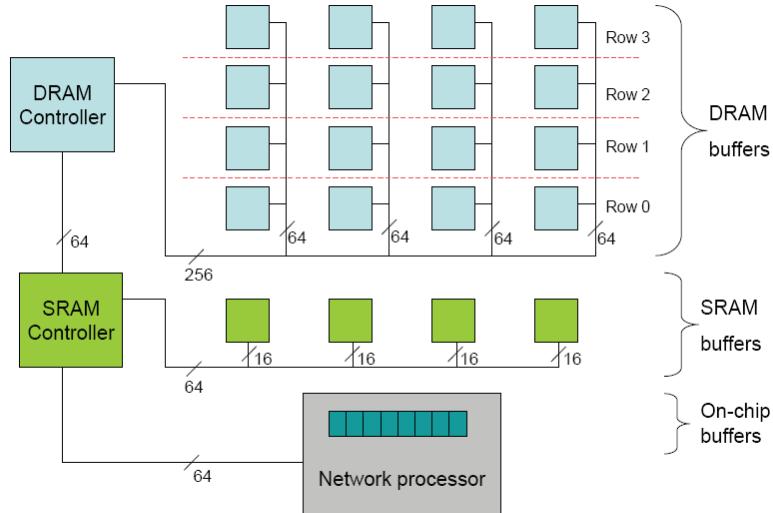


Figure 1. Buffer architecture in routers

As shown in Figure 1, it has three levels including on-chip buffers, off-chip cache (SRAM), and off-chip bulk memory (DRAM). The buffer memory can support multiple FIFO queues, and head and tail blocks of packets for each queue are moved between memory hierarchy levels as needed in a pipelined fashion. To meet the speed and latency requirements, a number of memory banks or chips are employed in parallel. Since DRAM is slower than SRAM, DRAM is accessed via a wider data bus. In Figure 1, DRAM organized as a 4×4 grid, with multiple chips within a row operating in parallel to increase data width, while each successive row adds to buffer depth.

The power consumed by DRAM is highly dependent on the frequency of read/write operations. To keep our energy model simple we approximate the DRAM as being in one of three states: active, idle, and sleep. The DRAM controller, which controls the entire DRAM buffers, is assumed to consume half the power of the entire DRAM memory. To save maximum energy, it is desirable to have the active buffer capacity track the actual queue occupancy, and to put to sleep any off-chip buffer memory that is not needed. However, the risk in doing so is that if a sudden burst of traffic arrives, there may not be sufficient time to activate buffer memory without dropping packets from this burst. In order to control how aggressively or conservatively we want to track the buffer occupancy, we introduce a novel algorithm which tries to make the total active buffer capacity at any time instant stay between the lower bound and upper bound of total buffer space. For example, in the extremely conservative setting of power conservation, we can set active buffers to maximal available buffers; in the aggressive settings, we can make the active buffer capacity track the exact queue occupancy.

5. Experiments and Performance Comparison

5.1. Experimental Setting

In this section, we conducted extensive experiments to evaluate the effectiveness of the ERM. The experiments are performed by using NS2 simulator, in which we integrated the routers

with our ERM mechanism. The network topology used in all the experiments is shown in Figure 2.

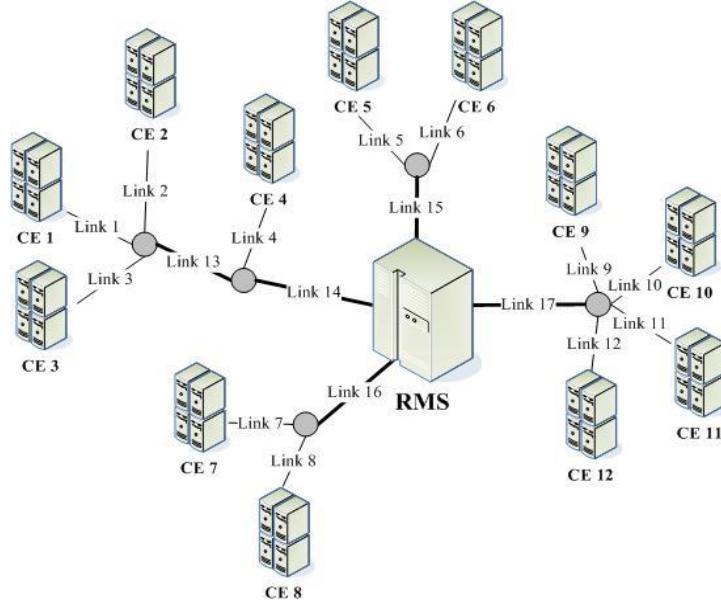


Figure 2. Topology of the experimental platform

As shown in Figure 1, the grid model consists of twelve computational elements (CE 1 ~ CE 12), each representing a high-performance cluster. The clusters are grouped into five groups by their geographical positions. Within each group, the clusters are connected by LAN (Link 1 ~ Link 12). Then, they are connected by WAN (Link 13 ~ Link 17) between groups. The failure of links follows exponential distribution with various parameters. According to the statistics in [17, 19, 21], the failures of LAN link are different from the WAN's. So, we set that the LAN's fault rate limited in [0.02, 0.04], and the WAN's fault rate limited in [0.04, 0.12].

In the simulations, the basic traffic is generated by using Lublin-Feitelson model [35], which is derived from the logs of real supercomputers. Each request is characterized by its arrival time T_a , resource demands R , execution time T_e . However, this basic workload can not meet the requirements of our simulation, because it lacks of latency requirement with respects with individual requests. So, we modify the basic workload by append each of them with latency constraint d , which is obtained as $d = T_a + g \cdot T_e$, where g is a random variable that uniformly distributed in [5.5, 10.5]. In addition, we all use FTP and Web traffic to investigate the performance of ERM, with aiming at assuring the traffic is highly dynamic just like real-world networks.

5.2. Comparison of Deadline-Miss Rate

In the first set of experiments, we compare the performance of ERM in term of deadline-miss rate with other three mechanism including CRM (Conventional Routing Mechanism) [36], EARTH (Energy-Aware service Rate Tuner Handling) [37], and ESIR (Energy Saving IP Routing) [38]. This experiment is divided into three groups by using different traffics (LF-traffic, FTP-traffic, Web-traffic). For each scenario, the experiment is conducted for four times. In the first three times, we turn off the ERM

and use the CRM, EARTH, ESIR as the default routing mechanism policy. Only in the fourth time, we turn on the ERM. All these four experiments use the same workload as their traffics. In order to make the system work in a stable state, we use CRM for the first ten minutes of the each simulation. The experimental results are shown in Figure 3.

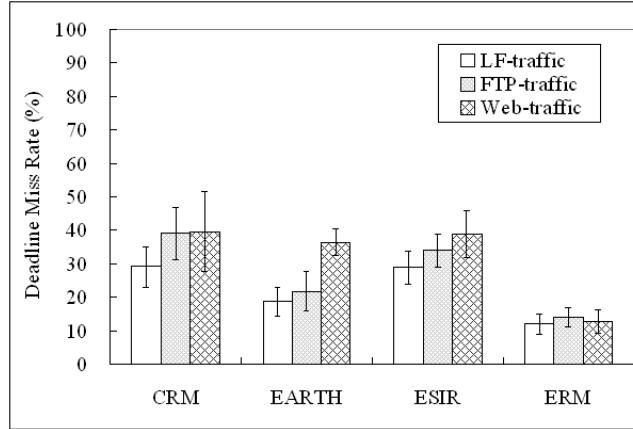


Figure 3. Deadline-miss rates with various routing mechanisms under different traffics

As we can see that, ERM is more effective to reduce deadline-miss rate comparing with CRM, EARTH, and ESIR in all experiments. As to the other three mechanisms, Web-traffic tends to result in higher deadline miss rate (DMR) than other workloads regardless of the underlying routing mechanisms. After analyzing the simulative logs in detail, we find that most of the requests in the Web-traffic have more stringent latency requirements than others. Consequently, its DMR metric is often highest. When using ERM, such a difference becomes very slight comparing with LF-traffic and FTP-traffic. This improvement contributes to the mechanism described in Section 4.1. In order to examine the detailed working of ERM, we record all the normalized transmission latency and DMR counts by different mechanisms (CRM is used as the baseline), which is shown in Table 1.

Table 1. Normalized transmission latency and DMR counts by different mechanisms

	Normalized transmission latency (Deadline-Miss Counts)		
	EARTH	ESIR	ERM
LF-traffic	0.85 (108)	0.88 (221)	0.68(256)
FTP-traffic	0.89 (166)	0.89 (256)	0.75 (99)
Web-traffic	0.96 (132)	0.94 (169)	0.71 (56)

It can be seen that transmission latency of ERM is about 68%~75% of the CRM, which improves about 20% comparing with ESIR and EARTH. The reason is that ESIR and EARTH are more aggressively to obtain energy saving than ERM. So, the network links are frequently working in SLEEP or D-SLEEP modes which significantly increase the latency. More importantly, the performance costs of switching operations itself also introduce a significantly latency. As mentioned in above sections, ERM relies on optional routing mechanism to saving energy consumption instead of frequently power

down routers. In this way, ERM outperforms over EARTH and ESIR in terms of DMR and transmission latency.

5.3. Comparison of Energy-Efficiency

To investigate the energy-efficiency of the proposed ERM, we introduce a novel measurement called Energy Consumption per Gb Transmission (EGT) to evaluate different routing mechanisms. As shown in Figure 2, each link capacity is set to 1Gbps, and packet size is set to randomly distributed in [0.1, 1.0] MB. The switching latency between ACTIVE and SLEEP is set as 0.01 ms and that between SLEEP and DSLEEP is set to 1 ms. The experimental results are shown in Figure 4.

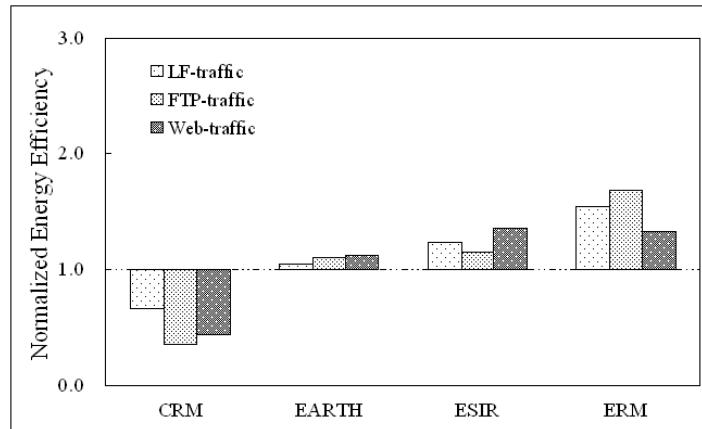


Figure 4. Normalized energy-efficiency by different mechanisms and traffics

As shown in Figure, CRM performs worst among the four mechanism in terms of normalized energy efficiency since it dose not adopt any energy saving optimization. As to EARTH, ESIR and our ERM, we notice that their energy efficiency depends on the traffics significantly. More specifically, for EARTH its energy efficiency is best when in presence of Web-traffic and worst for LF-traffic; for ESIR, its energy efficiency is best for Web-traffic and worst for FTP-traffic. As to our ERM, it performs best in presence of FTP-traffic and worst for Web-traffic. Even so, we can see that the general energy efficiency of ERM is significantly high than EARTH and ESIR (an exception occurs in case of Web-traffic).

By analyzing the procedure of simulations, we find that interval time between successive requests in LF-traffic seem too big. As a result, it often makes the underlying network resource underutilization, which in turn results in lower energy-efficiency. Such an experimental results is most significantly when using CRM that without any power saving mechanism. Although both EARTH and ESIR is incorporated with certain energy optimization mechanism, their approaches are quick different. For example, EARTH relies on workload prediction technology to adjust the service rate of routers. Therefore, if the traffic can be accurately predicted it will performs well. ESIR provides an online algorithm to adjust the link usage by controlling the router's working modes. In our experiments, Web-traffic is the most studied benchmark for investigating network performance, and its accessing pattern can be properly predicted by various kinds of intelligent technologies (*i.e.*, time series, fuzz logic and *etc*). So, both EARTH and ESIR can obtain better energy-efficiency for it.

As to our ERM, it relies on probability theory to calculate the latency guarantee and optional forwarding routing mechanism for optimizing router's energy consumption. Therefore, when the network traffic is more intensive its energy-efficiency will be better. In this experiment, FTP-traffic is the most intensive benchmark in terms of data transmission. In addition, it only put a very relaxing latency constrain on the packets. Therefore, ERM can easily select out those links that can satisfying the latency requirements of FTP-traffic's packets. When in presence of Web-traffic, ERM often have very stringent latency requirements which make it have fewer choices when selecting optional routings. This is the reason that ESIR outperforms ERM when in presence of Web-traffic about 3.4%.

6. Conclusion

In this work, we proposed an energy-aware routing mechanism for latency-sensitive traffics. It applies stochastic service model to calculate the latency-guarantee for any given network links and dynamically decides whether a link should be powered down and how long it should be kept in power saving mode. Extensive experiments are conducted to evaluate the effectiveness of the proposed mechanism, and the results indicate that it can provide better QoS performance for those latency-sensitive traffics with improved energy-efficiency. In the future, we are planning to take efforts on incorporating some adaptive admission controlling mechanisms into the routing mechanism for improving its energy-efficiency. In addition, we also plan to apply economic computing theory into the current system with aiming to provide better service for those budget-constrained network applications.

Acknowledgements

This work is supported by the Provincial Science & Technology plan project of Hunan (No.2012GK3075). Also, it is a project supported by Hunan Provincial Natural Science Foundation of China (No. 13JJ9022).

References

- [1] G. Grigoras, G. Cartina and E. C. Bobric, "Strategies for Power/Energy Saving in Distribution Networks", *Advances in Electrical and Computer Engineering*, vol. 10, no. 2, (2010), 61-64.
- [2] B. Tavli and W. B. Heinzelman, "QoS and Energy Efficiency in Network Wide Broadcasting: A MAC Layer Perspective", *Computer Communications*, vol. 30, no. 18, (2007), pp. 3705-3720.
- [3] L. Chiaraviglio, M. Mellia and F. Neri, "Minimizing ISP Network Energy Cost: Formulation and Solutions", *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, (2012), pp. 463-476.
- [4] A. Pathak and V. K. Prasanna, "Energy-Efficient Task Mapping for Data-Driven Sensor Network Macroprogramming", *IEEE Transactions on Computers*, vol. 59, no. 7, (2010), pp. 955-968.
- [5] C. Fragni and L. Costa, "ECO-ALOC: Energy-Efficient Resource Allocation for Cluster-based Software Routers", *Computer Networks*, vol. 56, no. 9, (2012), pp. 2249-2261.
- [6] R. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. W. Ayre and W. V. Sorin, "Evolution of WDM Optical IP Networks: A Cost and Energy Perspective", *IEEE Journal of Lightwave Technology*, vol. 27, no. 3, (2009), pp. 243-252.
- [7] S. Aleksic, "Analysis of Power Consumption in Future High-Capacity Network Nodes", *IEEE Journal of Optical Communications and Networking*, vol. 1, no. 3, (2009), pp. 245-258.
- [8] O. Tamm, "Scaling and Energy Efficiency in Next Generation Core Networks and Switches", In ECOC Sunday Workshop, Austria, (2009) September.
- [9] S. Newaz, A. Cuevas and G. M. Lee, "Improving Energy Saving in Time-Division Multiplexing Passive Optical Networks", *IEEE Internet Computing*, vol. 17, no. 1, (2013), pp. 23-31.

- [10] J. Rao and S. Biswas, "Network-assisted Sink Navigation for Distributed Data Gathering: Stability and Delay-energy Trade-offs", Computer Communications, vol. 33, no. 2, (2010), pp. 160-175.
- [11] C. H. Hsu and M. M. Hefeeda, "Broadcasting Video Streams Encoded With Arbitrary Bit Rates in Energy-Constrained Mobile TV Networks", IEEE/ACM Transactions on Networking, vol. 18, no. 3, (2010), pp. 681-694.
- [12] T. Mahmoodi, "Energy-aware Routing in the Cognitive Packet Network", Performance Evaluation, vol. 68, no. 4, (2011), pp. 338-346.
- [13] S. Kato, Y. Ishikawa and R. Rajkumar, "CPU Scheduling and Memory Management for Interactive Real-time Applications", Real-Time Systems, vol. 47, no. 5, (2011), pp. 454-488.
- [14] K. H. Kim, A. Beloglazov and R. Buyya, "Power-aware Provisioning of Virtual Machines for Real-time Cloud services", Concurrency and Computation-Practice & Experience, vol. 23, no. 13, (2011), pp. 1491-1505.
- [15] S. Iyer, R. R. Kompella and N. McKeown, "Designing Packet Buffer for Router Linecards", IEEE/ACM Transactions On Networking, vol. 16, no. 3, (2008), pp. 705-717.
- [16] A. A. Kist and A. Aldraho, "Dynamic Topologies for Sustainable and Energy Efficient Traffic Routing", Computer Networks, vol. 55, no. 9, (2011), pp. 2271-2288.
- [17] Y. Zhu, Y. Hu and M. B. Taylor, "Energy and Switch Area Optimizations for FPGA Global Routing Architectures", ACM Transactions on Design Automation of Electronic Systems, vol. 14, no. 1, (2009), pp. 1-34.
- [18] M. Andrews, A. F. Anta and L. Zhang, "Routing for Power Minimization in the Speed Scaling Model", IEEE/ACM Transactions on Networking, vol. 20, no. 1, (2012), pp. 285-294.
- [19] Micron-Technology-Inc., http://www.micron.com/support/dram/power_calc.html.
- [20] G. Epps. System Power Challenges, Cisco Research Seminar, (2006).
- [21] S. Nedevschi, L. Popa, G. Iannaccone and S. Ratnasamy, "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation", USENIX Symposium on Networked Systems Design and Implementation, (2008), pp. 323-336.
- [22] K. Kant, "Multi-state Power Management of Communication Links", International Conference on Communication Systems and Networks, (2011), pp. 1-10.
- [23] L. Chiaraviglio, M. Mellia and F. Neri, "Reducing Power Consumption in Backbone Networks", IEEE International Conference on Communications, (2009), pp. 1-6.
- [24] M. Zhang, C. Yi, B. Liu and B. Zhang, "GreenTE: power-aware traffic engineering", IEEE International Conference on Network Protocols, (2010), pp. 21-30.
- [25] S. Lee, P. Tseng and A. Chenc, "Link Weight Assignment and Loop-free Routing Table Update for Link State Routing Protocols in Energy-aware Internet", Future Generation Computer Systems, vol. 28, no. 2, (2011), pp. 437-445.
- [26] E. Amaldi, A. Capone, L. Gianoli and L. Mascetti, "Energy management in IP traffic engineering with shortest path routing", IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, (2011), pp. 1-6.
- [27] C. S. Raghavendra and S. Singh, "PAMAS: Power Aware Multi-access Protocol with Signaling for Ad Hoc networks", ACM SIGCOMM Computer Communication Review, vol. 28, no. 3, (1998), pp. 5-26.
- [28] W. Ye, J. Heidemann and D. Estrin, "An Energy-efficient MAC Protocol for Wireless Sensor Networks", IEEE International Conference of Computer and Communications, (2002), pp. 1567-1576.
- [29] T. Dam and K. Langendoen, "An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks", ACM International conference on Embedded networked sensor systems, (2003), pp. 171-180.
- [30] S. E. Lee and N. Bagherzadeh, "A High Level Power Model for Network-on-Chip (NoC) Router", Computers & Electrical Engineering, vol. 35, no. 6, (2009), pp. 837-845.
- [31] K. K. Ramachandran and B. Sikdar, "A Queuing Model for Evaluating the Transfer Latency of Peer-to-Peer Systems", IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 3, (2010), pp. 367-378.
- [32] A. Lombardo, C. Panarello and G. Schembra, "Achieving Energy Savings and QoS in Internet Access Routers", ACM SIGMETRICS Performance Evaluation Review, vol. 38, no. 3, (2010), pp. 76-80.
- [33] M. Barbera, A. Lombardo, C. Panarello and G. Schembra, "Active Window Management: an Efficient Gateway Mechanism for TCP traffic control", IEEE International Conference on Communications, (2007), pp. 1-9.
- [34] S. Khuller and B. Raghavachari, "Graph and Network Algorithms", ACM Computing Surveys, vol. 28, no. 1, (1996), pp. 43-45.
- [35] U. Lublin and D. G. Feitelson, "The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs", Journal of Parallel and Distributed Computing, vol. 63, no. 11, (2003), pp. 1105-1122.
- [36] V. Jacobson, "Congestion Avoidance and Control", ACM SIGCOMM, (1988), pp. 1-11.

- [37] C. Panarello, A. Lombardo, G. Schembra, L. Chiaravaglio and M. Mellia, "Energy Saving and Network Performance: a Trade-off Approach", ACM International Conference on Energy-Efficient Computing and Networking, (2010), pp. 41-50.
- [38] A Cianfrani, V Eramo, M Listanti, M Polverini and A. V. Vasilakos, "An OSPF-Integrated Routing Strategy for QoS-Aware Energy Saving in IP Backbone Networks", IEEE Transactions on Network and Service Management, vol. 9, no. 3, (2012), pp. 254-267.

Authors



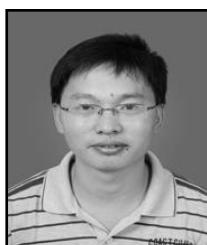
Xiao Peng

He received the Ph.D degree in computer science in CSU at 2010. He is currently an associate professor in the Hunan Institute of Engineering. His research interests include cloud computing, distributed resource management. He is a member of ACM and IEEE.



Qu Peixin

He is a lecturer in School of Information and Engineering, Henan Institute of Science and Technology. His research interesting is distributed computing, cloud computing, green computing.



Qu Xilong

He received his master degree and doctor degree in Xibei University. Now, he is associate professor of Hunan Institute of Engineering, master supervisor of Xiangtan University, the key young teacher of Hunan province, academic leader of computer application technology in Hunan Institute of Engineering. His research interesting is web service technology, information safety and networked manufacturing.

