Saving Energy and Improving TCP Throughput with Rate Adaptation in Ethernet

Y. Sinan Hanay, Wei Li, Russell Tessier and Tilman Wolf
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, USA
{hanay,wweili,tessier,wolf}@ecs.umass.edu

***Abstract*—Reducing the power consumption of network interfaces contributes to lowering the overall power needs of the compute and communication infrastructure. Most modern Ethernet interfaces can operate at one of several data rates. In this paper, we present Queue Length Based Rate Adaptation (QLBRA), which can dynamically adapt the link rate for Ethernet interfaces at runtime using existing Ethernet standards. An implementation of the proposed rate adaptation functionality is demonstrated at runtime on a NetFPGA platform. Our results show that the rate adaptation approach can achieve significant energy savings and at the same time improve the throughput of TCP traffic due to the effect of packet pacing.**

***Index Terms*—power consumption, link rate adaptation, packet pacing, TCP throughput**

I. Introduction

Energy consumption in information technology (IT) infrastructure is a pressing concern. The total amount of power consumed for computing and communication systems in data centers alone was estimated to be 61 billion kilowatt-hours in the U.S. [1]. Many system components in these data centers contribute to the overall power consumption, including servers, storage, networking equipment, power supplies, cooling, etc. Much work in computer system design in the past has focused on the main consumer of power within a computing system, the processor unit. However, there are many other components that contribute to power consumption. With an overall goal of making compute systems consume power proportional to their level of activity (i.e., "power-proportional") [2], it has become important to investigate the power consumption of all system components, including network interfaces (e.g., widely deployed Ethernet). At the same time, it is critical to understand the performance impact of energy-saving measures on the overall system

With these goals in mind, we present a novel Queue Length Based Rate Adaptation (QLBRA) for Ethernet. Our approach is based on the observation that Ethernet links can operate at different data rates (typically determined during auto-negotiation). With a small modification to the operation of an Ethernet interface, it is possible to support dynamic switching during operation of the link. To realize such an approach, there are several challenges that must be addressed to make such a system practical: (1) an interface cannot switch the data rate during transmission of a frame, thus we need a mechanism to determine at what rate to send; (2) the performance impact of slowing down link rates needs to be understood.

The Queue Length Based Rate Adaptation algorithm developed in our work uses a buffer to hold frames while waiting to transmit them. The length of that queue is used in determining at what rate to operate the link. This approach spaces out transmissions and thus can exploit the lower power requirements of using a lower data rate. The cost is a slightly increased delay that packets experience. However, as we show, this delay does not negatively affect the performance of the commonly used Transmission Control Protocol (TCP) due the effect of pacing [3].

The remainder of the paper is organized as follows. Section II discusses related work. Section III introduces our algorithm for adapting Ethernet link speed based on the queue length of the packet buffer. A system implementation on a FPGA platform is described in Section IV. Section V presents evaluation results on the energy savings and performance impact of our system. Section VI summarizes and concludes this paper.

II. Related Work

To save power in network interfaces, several different techniques have been proposed. IEEE standard 802.3az allows interfaces to sleep and "wake on arrival" when frames arrive; special care is taken to ensure that no transmissions are lost during the wake-up phase. Commercial network interfaces (e.g., [4]) also support a sleeping option. In addition, some interfaces can be configured to intentionally negotiate a lower data rate to save power [4]. However, this negotiation only occurs after a wake-up and thus the interface cannot adapt to changing traffic conditions.

Dynamic rate adaptation of the transmission link speed has been proposed in [5] and [6]. In [5], adapts the transmission rate to arbitrary values between the typical 10/100/1000 Mbps modes of Ethernet and thus is not realistic to implement. The work in [6] is similar to ours in that it uses a queue to determine transmission rates, but that work does not consider the impact of rate changes on upper layer protocols. Since we use a queue to buffer packets and determine the transmission rate based on queue length, we can show that our system has a similar effect as traffic pacing discussed in [3]. This pacing effect can improve the throughput performance of TCP connections as shown in the results in [3]. A rate adaptation algorithm supporting tri-mode operation was previously implemented in hardware in [7].

III. Queue Length Based Rate Adaptation

We propose to reduce the energy consumption in wired Ethernet interfaces by dynamically adapting the link rate at which the interface operates. The transmission rate is based on Queue Length Based Rate Adaptation (QLBRA), a simple algorithm that we describe in this section.

A. Energy Consumption in Ethernet

The power consumption in Ethernet is based on two components, the Ethernet media access controller (MAC) and Physical Layer Device (PHY). Approaches to saving power have focused on dynamic voltage scaling (to reduce power consumption in the logic blocks of MAC and PHY) as well as reducing overall device activity by initiating low-power or sleep modes. For example, a common interface from Intel provides low-power and sleep modes, a battery saver features for mobile devices that run on batteries (i.e., Auto Connect Battery Saver, Link Speed Battery Saver, Low Power Link Up) [4].

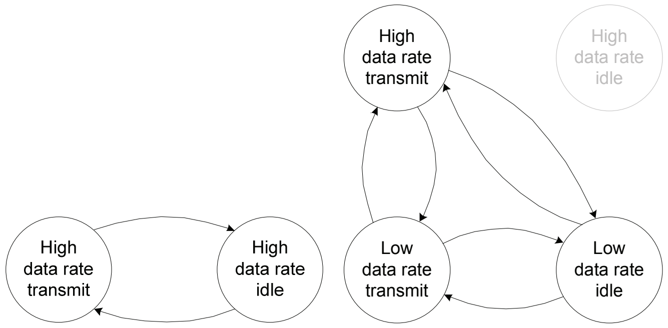
The existing approaches to reducing power consumption have two drawbacks:

- Sleep modes may be too drastic: Putting devices into sleep modes may have an impact on the state of a link (as perceived by the corresponding receiver) or may cause software support issues in operating systems. Also, specific protocols extensions may be necessary to recover from sleep mode (e.g., explicit wake-up). These approaches may lead to good power savings but may be unsuitable for some systems or traffic patterns where transitions into sleep modes are not warranted based on the potential power savings.
- Rate adaptation is too slow: Existing rate adaptation techniques change transmission rates at large timescales (e.g., once after wake-up). Since traffic patterns can change quickly during the operation of a network system, a faster adaptation technique is necessary to achieve power savings and to be able to provide good performance for any type of traffic.

To address these shortcomings, we propose to use Ethernet's ability to operate at different link rate to dynamically switch to the lowest (and least power-consuming) data rate that is suitable for traffic conditions. This solution leaves links in their active state (i.e., does not use sleep mode to reduce power) and can operate at the timescale of individual frame transmissions (i.e., and thus can adapt quickly). It is possible to use other power saving mechanisms (e.g., wake on arrival) in parallel to our proposed Queue Length Based Rate Adaptation.

B. Rate Adaptation

Rate adaptation can be implemented on Ethernet since practically all interfaces support multiple data rates (to be compatible with legacy devices). Most Gigabit Ethernet interfaces, for example, support data rates of 10 Mbps, 100 Mbps, and 1 Gbps. In a conventional setup, two communicating interfaces determine the highest common data rate and use that for communication. This "auto-negotiation" procedure is performed when the link becomes active and the selected data rate is used until the link is no longer active.



(a) Transmission States in Conventional Ethernet (b) Transmission States in QLBRA.

Fig. 1. Comparison of Transmission States in Conventional Ethernet and QLBRA.

In our system, we use this ability to support different data rates in a more dynamic fashion. Instead of committing to the highest data rate for the entire duration of connectivity, the transmitter can switch between the available link rates. In particular, the transmitter can select a lower data rate in order to save power. To illustrate the difference in operation, consider the state diagrams shown in Figure 1. For conventional Ethernet, the transmitter always uses the highest available data rate[1] and switches between transmit (i.e., active transmission) and idle (i.e., no transmission). In contrast, a system based on QLBRA uses its ability to switch to lower data rates when traffic permits and thus can save power. In particular, the high data rate idle state is avoided altogether (since for idle periods the system can always switch to the low data rate idle case).

The use of a lower data rate (for both transmit and idle states) directly translates into power savings. The different data rate modes in Ethernet are implemented different internal clocks in the PHY layer: for 10 Mbps, the clock is typically 2.5 MHz and Manchester Encoding is used for modulation in the PHY layer; for 100 Mbps, the clock is 25 MHz and Multi-Level Transmit (MLT-3) coding is used and the associated data path width is 4 bits; for 1 Gbps, the data path width is 8 bit and the clock frequency is 125 MHz. The use of higher clock rates implies more switching activities within a circuit and thus the use of more power. When limiting the use of a interface to lower data rates, portions of the circuit used for higher data rates can be turned of (e.g., using clock gating) to save power. (Table II in Section V below shows the experimental results for different data rates and activity states for a specific Ethernet interface. These results show that there is great potential in reducing power consumption by lowering the data rate of the interface.)

[1] For the discussion in this paper, we consider a link setup where there is a common "high" data rate and "low" data rate. In practice, this may translate to different specific data rates (e.g., 100 Mbps / 10 Mbps or 1 Gbps / 100 Mbps). There may also be cases with more than two possible data rates (e.g., 1 Gbps / 100 Mbps / 10 Mbps), which is discussed below. For simplicity, we limit most of our discussion to high and low data rates and let the reader extrapolate to other cases.

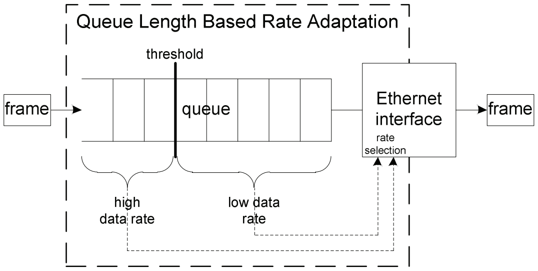


Fig. 2. Queue Length Based Rate Adaptation System Architecture.

C. Adaptation Algorithm

While it is intuitively clear that lowering the data rate in Ethernet can lead to a lower power consumption, an important question is how to do this specifically. In particular, it is important to determine the data rate for any given frame transmission such that the overall network performance does not suffer unduly.

To address this problem of selecting the interface data rate dynamically, we propose a Queue Length Based Rate Adaptation algorithm. This algorithm balances the need for simplicity (since it needs to be implemented efficiently in hardware at the interface) with the need for a good assessment of traffic conditions at runtime. The QLBRA system is illustrated in Figure 2 and consist of the following main components:

- Frame queue: If the arrival rate of frames does not exceed the maximum data rate of the interface, then there is no need for buffering in conventional Ethernet. However, the QLBRA system does not always transmit at full data rate. Therefore, it may happen that a frame is being transmitted at low data rate while other frames become ready for transmission. In this case, these frames need to be buffered. We use a simple frame queue with first-in-first-out (FIFO) ordering.
- Selection mechanism: The mechanism that selects the transmission data rate of the interface is very simple and only uses the current queue length as a metric for determining the outgoing data rate. If the queue length is below a certain threshold, then the low data rate is used. If the queue length is above the threshold, then the high data is used. Switching between data rates only occurs at frame boundaries to avoid that active transmissions are disrupted.
- Threshold parameter q_{th}: The threshold parameter determines the queue length cutoff at which the system switches from low data rate to high data rate. We discuss in Section V how different settings affect overall power savings and delay performance.

For duplex operation on a network link, we assume that the transmitting interface, which knows how many frames are ready for transmission, determines the link speed according to the above algorithm. Thus, for duplex operation, each transmitter can make this decision independently.

D. Buffer Requirements and Delay

One key question in the context of QLBRA is the amount of buffer space that is necessary when implementing the system. Clearly there has to be at least enough buffer space to accommodate as much data as specified by the threshold (expressed in bytes rather than frames). However, the threshold can be exceeded in certain traffic scenarios. The worst case example is that a low data rate transmission of a maximum size frame is started with a queue length just below the threshold. If during this transmission many frames arrive, then the queue length exceeds the threshold considerably. However, since the incoming rate of frames is bounded by the high transmission rate, $rate_{high}$ and the length of frames is bounded by the maximum frame size, $frame_{max}$, we can provided an overall bound on the queue size, q_{max}. This bound is:

$$q_{max} = q_{th} + \frac{rate_{high}}{rate_{low}} \cdot frame_{max} \tag{1}$$

In a system where the high data rate is 10× that of the low data rate (e.g., 100 Mbps and 10 Mbps), the maximum queue length is the threshold plus 10 maximum-length frames. With the small threshold values that suffice based on the results provided in Section V, clearly this amount of buffering can be implemented easily in a system.

One other concern for QLBRA is that using lower data rates for transmission (and the associated queueing) increases the delay experienced by frames traversing the system. For low-latency, local area systems, this delay can be of concern. However, as we see from the results in Section V, the delays incurred by QLBRA are considerably less than typical propagation delays on long-haul links. Thus, these delay does not matter for traffic that crosses these distances.

E. Pacing Effect

Another benefit of the QLBRA system comes from its implicit packet pacing effect. Pacing of TCP traffic has been shown to increase throughput under certain traffic conditions. One specific such pacing approach is Queue Length Based Pacing (QLBP) [3], which also uses the queue length of the transmission buffer as an indicator to determine, in this case, pacing delays. QLBP has been shown to improve TCP throughput, especially in network with routers with small buffers (e.g., optical packet-switched networks).

Our QLBRA can be thought of as a simplified version of QLBP. In QLBP, a rate controller sets the transmission rate (and thus the pacing delay) at any continuous value between the minimum and the maximum rate. In contrast, QLBRA requires rates to be exactly one of the available transmission rates of the interface. In Section V, we show that despite being simple than QLBP, QLBRA still achieves similar pacing properties.

IV. System Implementation

We have implemented our QLBRA system on NetFPGA, an open FPGA platform for network research [8]. The implementation follows the algorithm described in Section III. A Xilinx

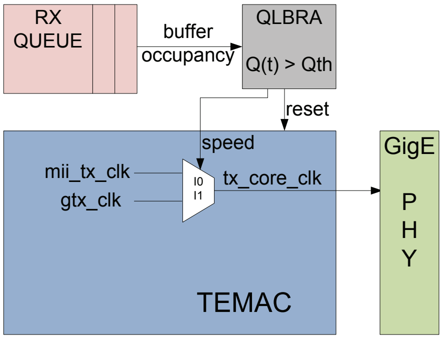


Fig. 3. System Diagram of QLBRA MAC Prototype.

TABLE I
RESOURCE USAGE OF QLBRA ON NETFPGA. PERCENTAGES REFER TO RESOURCE USAGE RELATIVE TO TOTAL AMOUNT OF RESOURCES ON VIRTEX II.

Resource Type	NetFPGA Reference Router	Reference Router with QLBRA
Slices	16,431 (34%)	16,435 (34%)
Block RAMs	106 (45%)	106 (45%)
DCMs	6 (75%)	7 (87%)
BUFGMUX	8 (50%)	10 (62%)

TEMAC core was used inside the NetFPGA's Virtex II FPGA as the Ethernet MAC. The block diagram of the prototype implementation is shown in Figure 3. A monitor implemented in FPGA logic dynamically determines if the on-FPGA buffer occupancy is filled beyond a prespecified threshold, q_{th}. If so, the operation of the TEMAC core is changed to high-speed mode. TEMAC speed changes require the core to be held in reset for three 62 MHz clock cycles. This overhead is small enough that it does not affect overall throughput performance.

Table I shows the resource usage in the Virtex II device. The results shows that QLBRA introduces only a small area overhead and requires one extra digital clock manager (DCM).

The media independent interface (MII) is an interface between MAC and PHY that supports 10/100 Mbps modes. Gigabit MII (GMII) is the extended version that supports 1 Gbps. There are two possible ways to configure MAC to run at 10/100/1000. One is to use GMII/MII, the other is use of Reduced GMII (RGMII). In GMII/MII for 100 Mbps, the clock is provided from PHY and for 1 Gbps the clock is provided by MAC. However, with RGMII both clocks are provided by MAC, so instead we use RGMII for simplicity. Additionally, the MII interface is used for 10 Mbps and 100 Mbps transfers and the Gigabit MII is used for 1 Gbps transfers. The MII clock is externally provided by the PHY, while for the GMII the MAC provides the transmit clock to the physical layers. Since the MII data width is 4 bits and the GMII data width is 8 bits, the transmit clock enable signal is always asserted high for MII while for GMII it is asserted high on alternating cycles [9]. In our design, only 100 Mbps and 1 Gbps speeds are used and clock enables are not used. The MAC provides the transmit clock to the respective PHY.

As shown in Figure 3, the QLBRA module synchronously monitors the receive queue (RX), and when the size of the receive queue exceeds q_{th}, QLBRA signals the MAC to switch to high speed mode during a system-wide reset of three cycles. The gtx_clk is a 125 MHz clock provided by the FPGA to allow for 1 Gbps transfers. The gtx_clk is divided by 5 by a digital clock manager (DCM) to form the 25 MHz mii_tx_clk. This mii_tx_clk is used for 100 Mbps mode. The multiplexer in Figure 3 is a BUFGMUX, which facilitates clock switching with a minimum of glitching. When the speed changes, the transmitter send a control frame to notify the receiver of a speed change.

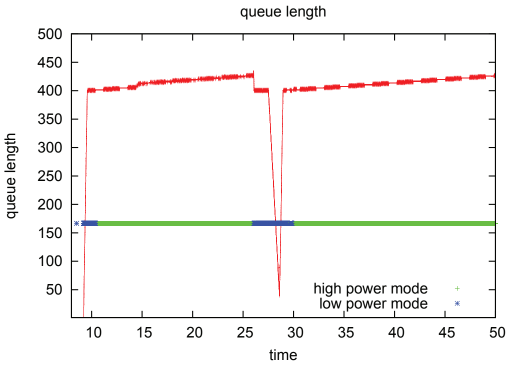


Fig. 4. Queue Length Based Rate Adaptation Operation of Prototype System.

Figure 4 demonstrates the operation of QLBRA. In this example q_{th} is set to 400 packets, and depending on the queue length, the operation modes are adjusted. As expected, whenever the queue length goes beyond the threshold of 400 packets, the MAC uses the high data rate mode.

V. EVALUATION

Based on the prototype implementation, we have evaluated the QLBRA system and its ability to reduce power consumption on network interfaces and increase the throughput of TCP traffic.

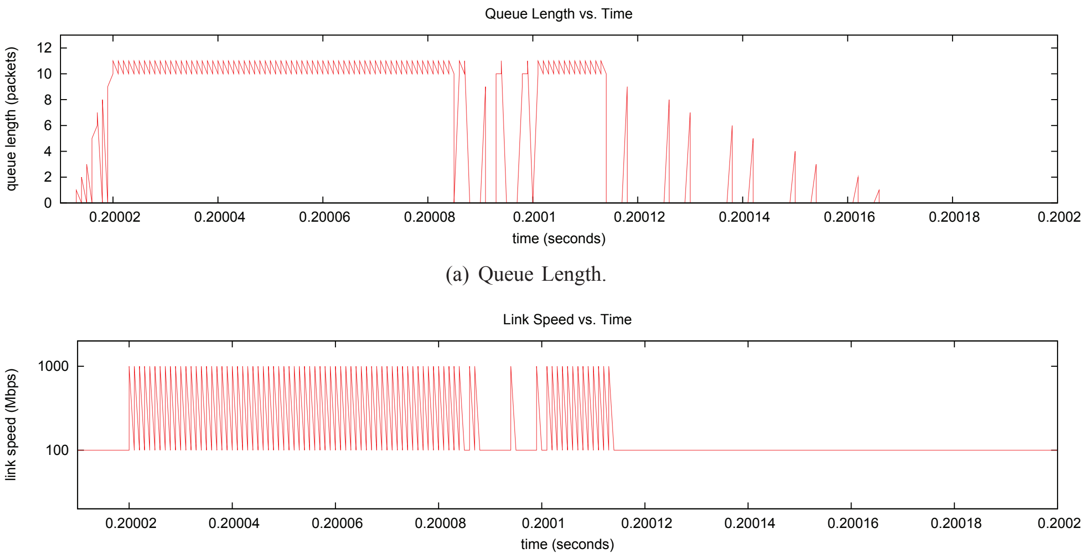
A. Experimental Setup

To achieve a quantitative evaluation of our system, we have set up experiments where we use the ns-2 simulator [10] to determine how the system operates under different traffic conditions and threshold settings. From these simulations, we obtain information about when the system operates in which state. This information is then combined with the power estimations based on our FPGA prototype to provide power and energy results. The simulations also provide information about the frame delays and TCP performance. Except for experiments involving TCP, the simulations use CAIDA traffic traces from the Equinix-Chicago link [11], an OC-192 link operating at 10 Gbps. For our experiments, the data rate was scaled to one tenth to allow for use with link rates of 1 Gbps ("high") and 100 Mbps ("low").

Power consumption in all digital circuits, including the TEMAC core circuits used in this experimentation, is a combination of static and dynamic power. Static power dissipation is generally the result of transistor leakage currents in CMOS

TABLE III
POWER AND ENERGY CONSUMPTION.

	Trace 1 (load=3.3%)			Trace 2 (load=72%)		
Interface	Delay (μs)	Total energy (J)	Avg. power (W)	Delay (μs)	Total energy (J)	Avg. power (W)
Static rate	0	174.60	2.91	0	225.32	3.75
QLBRA ($q_{th} = 3$)	1.63	112.07	1.86	2.49	192.79	3.21
QLBRA ($q_{th} = 5$)	1.94	111.14	1.85	4.16	192.75	3.21
QLBRA ($q_{th} = 10$)	2.19	111.62	1.86	8.47	192.67	3.21
QLBRA ($q_{th} = 20$)	2.29	111.33	1.85	16.78	192.02	3.20



(a) Queue Length.

(b) Transmission Rate.

Fig. 5. Comparison of Queue Length and Rate Change over Time for $q_{th} = 10$.

TABLE II
POWER CONSUMPTION FOR DIFFERENT INTERFACE STATES OF TEMAC WITH INTEL 82567.

Interface speed	Power consumption	
	Active (W)	Idle (W)
100 Mbps	2.50	1.83
1 Gbps	4.10	2.86

circuits. This dissipation is directly related to the supply voltage and is generally independent of circuit activity. In this work, the Xilinx XPower tool is used to determine the dynamic power of TEMAC core by specifying the toggling percentage of core inputs. The XPower tool automatically uses the transition density model to determine internal circuit logic toggle percentages. For PHY power consumption estimation, reported power information for the Intel 82567 tri-mode GbE PHY Transceiver was used. Although NetFPGA uses a Broadcom BCM5464SR PHY, its datasheet is available only through NDA, necessitating power estimation using a similar PHY. Power estimates are shown in Table II.

B. Energy Reduction

For our energy saving calculations, we excite the QLBRA system and a static Ethernet interface with the same traffic. The traffic used in these simulations come from two different CAIDA traces, which are 60 seconds long. One of the traces is lightly loaded ($\sim$ 4 million packets), one of them is heavily load ($\sim$ 65 million packets). Table III shows the energy savings for different setting of threshold parameters. We see from the table that for a delay of a few micro seconds, it is possible to save 36% of energy when link is lightly loaded and 14% when link is heavily loaded. This level of power saving is considerable, especially since it only requires a simple change in operation of the Ethernet interface.

With a threshold of $q_{th} = 10$, Figure 5(a) shows the queue length evolution for the lightly loaded trace for a few milliseconds. Figure 5(b) shows the corresponding rate change over time.

C. Packet Delay

Figure 6 shows the added mean queueing delay for different threshold sizes, q_{th}. Setting a small threshold gives lower delay, but also less energy savings, while a bigger threshold gives more energy savings at the cost of longer queueing delays. The energy consumption versus threshold is shown in Figure 7.

D. TCP Performance

We also evaluate the performance of this QLBRA in the context of TCP. As discussed above, QLBRA acts as a pacer. We use a dumbbell topology as it is common in previous work,

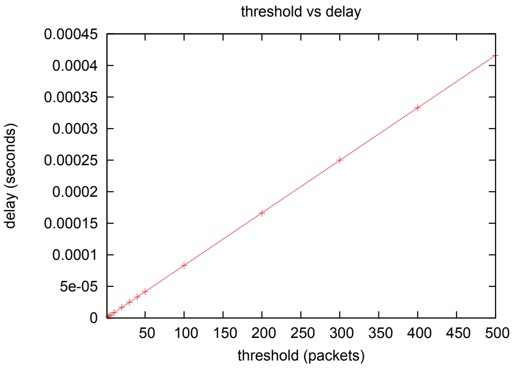


Fig. 6. Queue Length Based Rate Adaptation Delay vs Threshold.

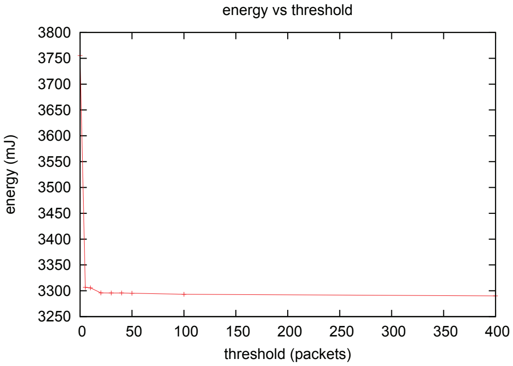


Fig. 7. Queue Length Based Rate Adaptation Energy vs Threshold.

and pacers are implemented at the outgoing links of access routers. There are 10 traffic sources connected to 4 access routers. Each traffic source sends 10 traffic flows.

Figure 8 shows the normalized aggregate throughput of 100 flows versus various buffer sizes. Packet pacing achieves higher throughput especially in small buffer networks. The figure shows that for buffer sizes of 80 packets or less, QLBRA gives better performance than static interface. Moreover, QLBRA's performance is close to that of QLBP, and both of them are better than static interface case. This results show that in certain situations (e.g., in small-buffer networks), *QLBRA can reduce energy consumption while at the same time providing TCP throughput*.

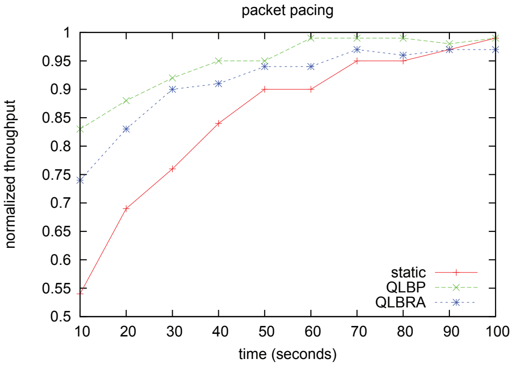


Fig. 8. Pacing Effect of QLBRA.

VI. Summary and Conclusions

Energy consumption in network interfaces contributes to the overall energy consumption of computer and communication infrastructure. Since wired Ethernet interfaces are widely deployed, it is important to find a way to reduce their power consumption while still adhering to existing standards. In this work, we propose Queue Length Based Rate Adaptation, which uses the ability to dynamically switch between data rates in Ethernet. Using a small buffer and determining the data rate based on queue length enables QLBRA to reduce power and energy consumption of 14%–36%. While QLBRA introduces some delay, this delay is only around 2 μs – 17 μs and thus does not reduce throughput performance. Instead, for some traffic scenarios, QLBRA acts as a traffic pacer that actually improves TCP throughput. This result can provide a basis for changing the misconception that saving power necessarily is associated with a decrease in performance. In particular, since our approach does not require changes to the standards of Ethernet, QLBRA has the potential of becoming widely deployed in practice.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0721790.

References

[1] *Report to Congress on Server and Data Center Energy Efficiency – Public Law 109-431*, U.S. Environmental Protection Agency, Aug. 2007.

[2] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, pp. 33–37, Dec. 2007.

[3] Y. Cai, Y. S. Hanay, and T. Wolf, "Practical packet pacing in small-buffer networks," in *Proc. of IEEE International Conference on Communications (ICC)*, Dresden, Germany, Jun. 2009.

[4] *82567 GbE physical layer transceiver (PHY)*, Intel Corporation, 2009, datasheet.

[5] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proc. of the Fifth USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, Apr. 2008, pp. 323–336.

[6] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (ALR)," *IEEE Transactions on Computers*, vol. 57, pp. 448–461, Apr. 2008.

[7] B. Zhang, K. Sabhanatarajan, A. Gordon-Ross, and A. D. George, "Real-time performance analysis of adaptive link rate," in *Proc. of IEEE Conference on Local Computer Networks (LCN)*, Montreal, Canada, Oct. 2008, pp. 282–288.

[8] J. W. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo, "NetFPGA–an open platform for gigabit-rate network switching and routing," in *MSE '07: Proceedings of the 2007 IEEE International Conference on Microelectronic Systems Education*, San Diego, CA, Jun. 2007, pp. 160–161.

[9] *LogiCORE IP Tri-Mode Ethernet MAC v4.5*, Xilinx Corporation, 2011, xilinx User's Guide 138.

[10] *The Network Simulator - ns-2*, LBNL, Xerox PARC, UCB, and USC/ISI, http://www.isi.edu/nsnam/ns/.

[11] k. claffy, D. Andersen, and P. Hick, "The CAIDA anonymized 2011 Internet traces dataset – February," http://www.caida.org/data/passive/passive_2011_dataset.xml.