Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)

Chamara Gunaratne, *Student Member*, *IEEE*, Ken Christensen, *Senior Member*, *IEEE*, Bruce Nordman, and Stephen Suen

Abstract—The rapidly increasing energy consumption by computing and communications equipment is a significant economic and environmental problem that needs to be addressed. Ethernet network interface controllers (NICs) in the US alone consume hundreds of millions of US dollars in electricity per year. Most Ethernet links are underutilized and link energy consumption can be reduced by operating at a lower data rate. In this paper, we investigate Adaptive Link Rate (ALR) as a means of reducing the energy consumption of a typical Ethernet link by adaptively varying the link data rate in response to utilization. Policies to determine when to change the link data rate are studied. Simple policies that use output buffer queue length thresholds and fine-grain utilization monitoring are shown to be effective. A Markov model of a state-dependent service rate queue with rate transitions only at service completion is used to evaluate the performance of ALR with respect to the mean packet delay, the time spent in an energy-saving low link data rate, and the oscillation of link data rates. Simulation experiments using actual and synthetic traffic traces show that an Ethernet link with ALR can operate at a lower data rate for over 80 percent of the time, yielding significant energy savings with only a very small increase in packet delay.

Index Terms—Power management, energy-aware systems, local area networks, Ethernet, Energy Efficient Ethernet.

1 Introduction

The Internet is rapidly becoming a major consumer of electricity with measurable economic and environmental impact. The hubs, switches, and routers that comprise the core of the Internet consumed an estimated 6.15 TWh/yr in 1999 [31] (with an expected increase of 1 TWh/yr by 2005). One TWh/yr corresponds to $85 million at $0.085 per kWh [35] and about 0.75 million tons of CO_2 [3]. Projections of growth are less clear, but it appears that electronic equipment (most of it connected to the Internet) is the fastest growing consumer of electricity. In 2000, it was estimated that 9 percent of the commercial sector electricity consumption was due to electronic office and telecommunications equipment [31]. In addition to the network equipment at the core of the Internet are all of the desktop PCs and new commercial and residential devices that connect to the Internet via Ethernet links. It is estimated that the energy consumption of the Ethernet network interface controllers (NICs) in desktop PCs and other network edge devices in the US was approximately 5.3 TWh/yr in 2005 [26]. The energy used by Ethernet links is growing rapidly as the default link data rate for desktop PCs and other network edge devices increases from 100 Mbps to 1 Gbps (and eventually to 10 Gbps), and the overall number of Ethernet-connected devices also increases.

Measurements have shown that 1 Gbps Ethernet links consume about 4 W more than 100 Mbps links [16]. A 10 Gbps Ethernet link may consume more, from 10 to 20 W [16]. We have found that idle and fully utilized Ethernet links consume about the same amount of power. That is, Ethernet power consumption is independent of link utilization. Measurements show that the average utilization of desktop Ethernet links is mostly in the range of 1 percent to 5 percent [5], [28]. Thus, there is opportunity for significant energy savings with user-imperceptible impact to packet delay by operating links at a lower data rate during low utilization periods. We seek to improve Ethernet so that energy use is proportional to link utilization. One means to achieve this is to adaptively vary the link rate to match the offered load or utilization. Adaptive Link Rate (ALR) was proposed by Nordman and Christensen at an IEEE 802.3 tutorial in July 2005 as a means of automatically switching the data rate of a full-duplex Ethernet link to match link utilization [26]. ALR was first described in [16] by Gunaratne et al. ALR is intended to use existing Ethernet data rates (that is, 10 Mbps, 100 Mbps, 1 Gbps, and 10 Gbps) and is intended primarily for edge links. ALR *mechanisms* determine how the link data rate is switched and ALR *policies* determine when to switch the link data rate. A good policy should maximize the time spent in a low data rate while minimizing increased packet delay. Thus, the key performance trade-off is packet delay versus energy savings. Energy savings are achieved by operating at a low link data rate (for example, 100 Mbps instead of 1 Gbps). In this paper, we present a system design for ALR and investigate ALR policies using analytical and simulation modeling.

The remainder of this paper (which builds upon and summarizes previously submitted work by the same

- *C. Gunaratne and K. Christensen are with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620. E-mail: {pgunarat, christen}@cse.usf.edu.*
- *B. Nordman is with the Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720. E-mail: bnordman@lbl.gov.*
- *S. Suen is with the Department of Mathematics and Statistics, University of South Florida, Tampa, FL 33620. E-mail: suen@math.usf.edu.*

Manuscript received 10 Jan. 2007; revised 3 July 2007; accepted 5 Oct. 2007; published online 15 Oct. 2007.
Recommended for acceptance by V. Barbosa.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0012-0107.
Digital Object Identifier no. 10.1109/TC.2007.70836.

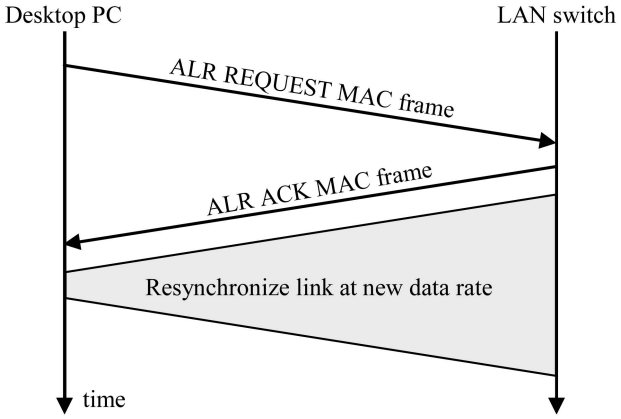

Fig. 1. Timing diagram for an ALR MAC frame handshake mechanism.

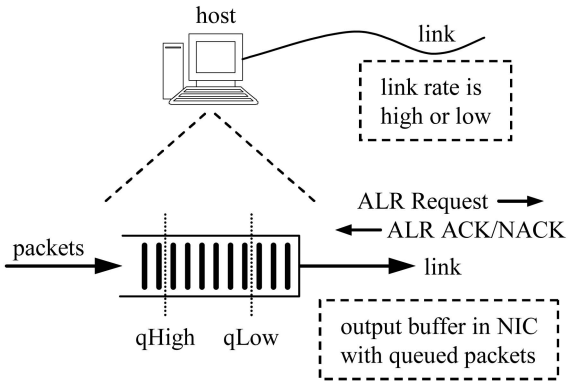

Fig. 2. Example NIC output buffer with high and low thresholds.

authors in [15] and [17]) is organized as follows: Section 2 describes the ALR MAC frame handshake mechanism and the dual-threshold, utilization-threshold, and time-out-threshold policies. Section 3 develops a Markov model for the dual-threshold policy. Section 3 also presents numerical results and analyzes link rate oscillations for the three ALR policies. The sampling period needed for the utilization-threshold policy is investigated in Section 4. Section 5 presents a simulation model-based performance evaluation for representative link configurations given both trace and synthetic traffic as input. Potential energy savings given widespread deployment of ALR are estimated in Section 6. Related work is reviewed in Section 7. Finally, Section 8 summarizes the paper and discusses future work.

2 Ethernet Adaptive Link Rate

The key challenges in realizing ALR are 1) defining a mechanism for quickly switching link data rates and 2) creating a policy to change the link data to maximize energy savings (that is, maximize time spent in a low link data rate) without significantly increasing the packet delay.

2.1 ALR MAC Frame Handshake Mechanism

A fast mechanism for initiating and agreeing upon a link data rate change is necessary. Either end of a link (for example, the NIC in a desktop PC or switch port in a LAN switch) must be able to initiate a request to change the rate. ALR can only operate within the bounds of an advertised capability of both ends of a link. It is necessary at link establishment to negotiate capabilities, including the support of ALR at both ends and the possible data rates that the link partner NICs have in common. Existing IEEE 802.3 Auto-Negotiation [21, Clause 28] can be used at link establishment for capability negotiation. Auto-Negotiation could also be used to switch data rates during link operation, but this would require a minimum of 256 ms (and, in reality, typically several seconds). Such a long handshake time would make ALR infeasible due to its effect on increasing packet delay. A faster handshake can be implemented using Ethernet MAC frames. A two-way handshake could be implemented as follows:

- The end of the link that determines a need to increase or decrease its data rate requests a data rate change using an ALR REQUEST MAC frame. The request could be "goto low" or "goto high" data rate.
- The receiving link partner acknowledges the data rate change request with either an ALR ACK (agrees to change the data rate) or ALR NACK (does not agree) MAC frame.

Fig. 1 shows a MAC frame handshake followed by a link resynchronization.

An ALR ACK response would trigger a link data rate switch (rate transition) and link resynchronization. The total time for the handshake plus resynchronization, T_{switch}, could likely be less than 100 microseconds for 1 Gbps Ethernet. This is based on a reasonable 10,000 clock cycles needed for link resynchronization. For 10 Gbps, more work needs to be done to understand if link retraining needs to be done every time a link is resynchronized or only at link establishment. The duration of T_{switch} is critical to the performance of ALR.

2.2 ALR Dual-Threshold Policy

The simplest ALR policy is based on output buffer occupancy, or queue length, threshold crossing. Fig. 2 shows an NIC output buffer with high (qHigh) and low (qLow) queue thresholds. Two thresholds are used to introduce hysteresis into the system and prevent a trivial oscillation between rates. The dual-threshold policy is shown in the finite-state machine (FSM) in Fig. 3, where the output buffer queue thresholds are qLow and qHigh. Fig. 4 defines the FSM timers, system parameters, and internal variables for this FSM and also for a subsequent FSM. The states HIGH and LOW correspond to high and low link data rates. Timers are expired when not timing and, when reset, they begin to count down to their expired condition. The actions to be taken if link resynchronization fails are not shown. It is assumed that one side starts (for example, at link initialization) with reqLow as true and the other side with it as false and both sides always start in the HIGH state.

If the output buffer queue length in a NIC or switch port exceeds qHigh, then the data rate must be transitioned to high, as shown in transition 8 in the FSM in Fig. 3. The only allowable response from the other side is an ACK. That is, the other side cannot disagree with a request to increase the link data rate. If the output queue decreases below qLow, then the data rate can be reduced to low if both sides agree. This is shown in transitions 1 and 2. If the other side does not agree and returns a NACK, as shown in transition 3, it is

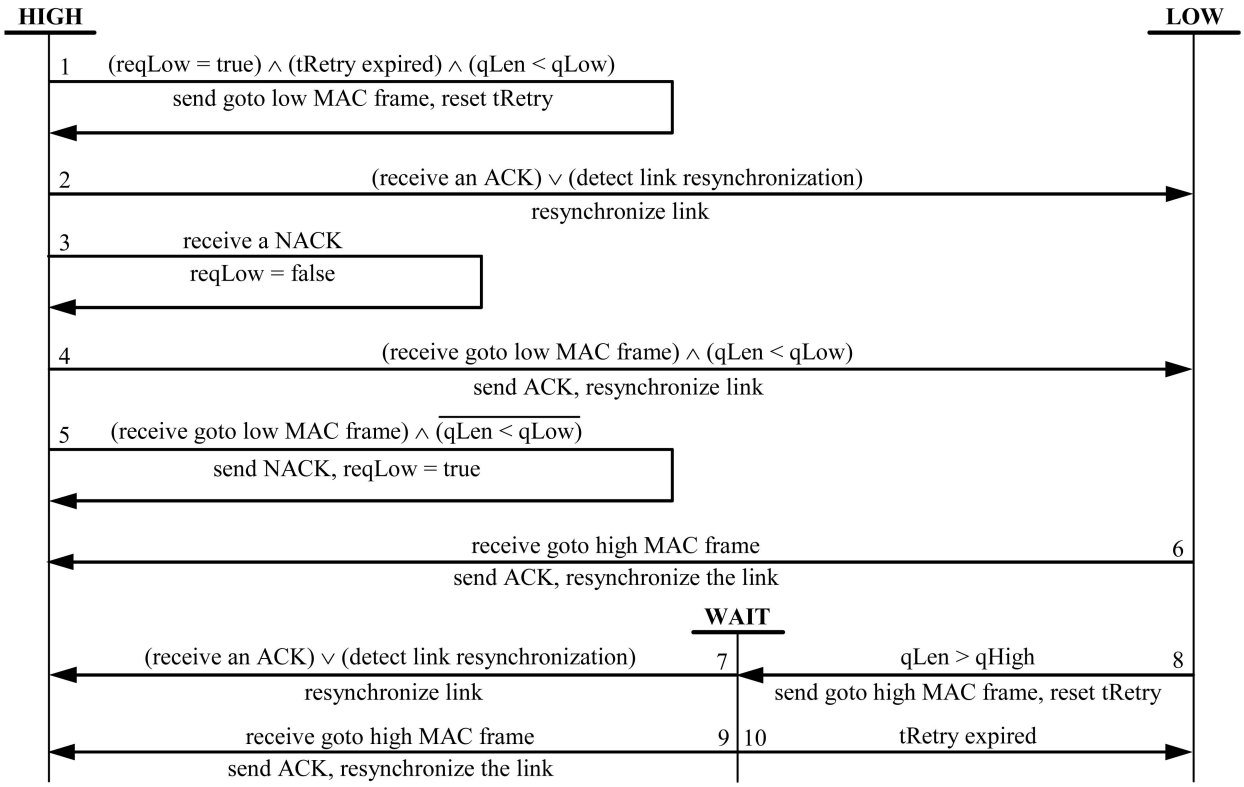

Fig. 3. FSM for the ALR dual-threshold policy.

now this other side that must request the low data rate when its queue length is below its qLow threshold. Transitions 4 and 5 show the other side requesting to reduce the rate to low and this side agreeing by sending an ACK in transition 4 or not agreeing by sending a NACK in transition 5. The internal variable reqLow in the FSM is used to prevent one side from repeatedly requesting to reduce the data rate when the other side cannot agree with the request (for example, if one side has high utilization and the other side is idle).

2.3 ALR Utilization-Threshold Policy

The dual-threshold policy can oscillate between data rates given smooth traffic input or bursty traffic with long duration bursts causing increased response time and variability. The oscillation of link rates will occur if the packet arrival rate at the low link data rate is high enough to cause a high threshold crossing (qHigh) at the low data rate but not high enough to maintain the queue length above the low threshold (qLow) at the high data rate. To study the problem of link rate oscillation, we conducted an experiment for a simulated NIC with link data rates of 100 Mbps and 1 Gbps given Poisson (smooth) arrivals of packets of constant length of 1,500 bytes. This system (without ALR) can be trivially modeled as an M/D/1 queue. Fig. 5 shows the utilization at 1 Gbps versus the mean packet delay for an M/D/1 queue and for a simulated M/D/1 queue with the dual-threshold ALR policy. The two dashed-line curves in Fig. 5 show the mean packet delay for M/D/1. As the packet arrival rate (the utilization) increases when in the low link data rate, the packet delay (that is, the queuing

tRetry	Timer for handshake retry (ACK, NACK lost)
qLow	Queue low threshold in bytes
qHigh	Queue high threshold in bytes
qLen	Queue length in bytes
reqLow	Request low rate flag (this side to request)

(a)

tUtil	Timer for utilization measurement interval
uBytes	Utilization (bytes transmitted in last interval)
uThresh	Utilization threshold in bytes
txLen	Transmitted packet length in bytes
txSum	Transmitted bytes accumulator variable

(b)

Fig. 4. ALR variables, timers, and parameters. (a) For the FSM in Fig. 3. (b) For the FSM in Fig. 6.

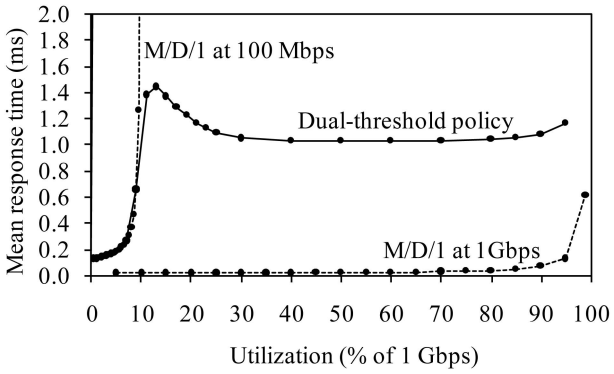

Fig. 5. ALR dual-threshold policy and M/D/1.

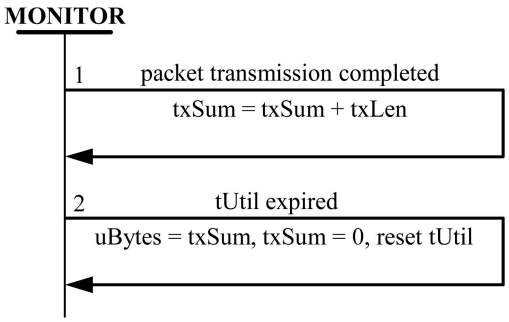

Fig. 6. FSM for utilization monitoring.

delay) also increases and will reach an unacceptable level. Thus, at some utilization level less than 10 percent of 1 Gbps, the data rate should be switched from 100 Mbps to 1 Gbps and remain in the 1 Gbps high link data rate. For the simulated M/D/1 queue, we used $\mathrm{qLow} = 0$ Kbytes, $\mathrm{qHigh} = 32$ Kbytes, and $T_{switch} = 1$ ms, corresponding to what we believe is a reasonable output buffer size [2] and rate-switching time for an Ethernet NIC. We assumed that all ALR data rate switch requests are always ACKed. The mean packet delay for the ALR-controlled queue is the solid-line curve in Fig. 5. It can be seen that mean packet delay increases similarly to the M/D/1 curve for 100 Mbps, but does not decline once the arrival rate (utilization) is greater than 100 Mbps (which occurs at 10 percent utilization at 1 Gbps). This is due to the dual-threshold policy causing the link data rate to oscillate between 100 Mbps and 1 Gbps and incur significant delays from the 1 ms rate switching time (T_{switch}). Oscillation is studied further in Section 3.3.

To prevent oscillation, a utilization-threshold policy was developed. If link utilization is explicitly monitored and used in the decision to transition between data rates, the effect of oscillations on packet delay, seen in Fig. 5, can be reduced or eliminated. The FSM in Fig. 6 shows the monitoring for link utilization by counting the number of bytes transmitted during a time period (the timer, thresholds, and variables are defined in Fig. 4). Utilization monitoring is based on counting the bytes sent in an interval tUtil. An explicit utilization test is added to transitions 1, 4, and 5 in the FSM in Fig. 3. The condition $(\mathrm{qLen} < \mathrm{qLow})$ is replaced with $((\mathrm{qLen} < \mathrm{qLow}) \wedge (\mathrm{uBytes} < \mathrm{uThresh}))$. These changes to the FSM in Fig. 3 and the new FSM in Fig. 6 constitute the utilization-threshold policy.

2.4 ALR Time-Out-Threshold Policy

Counting the number of bytes transmitted within a time period requires additional accumulators and registers that could increase the complexity of an ALR-capable NIC. Therefore, we also investigated a heuristic policy—called the ALR time-out-threshold policy—to maintain the link at a high data rate for a predetermined period of time following a switch from a low to a high link data rate. For the ALR time-out-threshold policy, two new timers are defined: 1) tMinHigh for the minimum time for the link to stay in the high data rate and 2) tMinLow for the minimum time the link should stay in the low data rate before switching to the high data rate. The timer tMinHigh is reset and restarted upon switching to the high rate and the link data rate is maintained at the high rate until the timer expires, irrespective of the queue length. When tMinHigh has expired, if the queue length is below the qLow threshold, the link data rate is switched to the low rate. The timer tMinLow is always reset and restarted upon switching to the low data rate. Switching to the high rate is triggered by a queue threshold crossing of the qHigh threshold even if tMinLow has not expired. The policy can be made adaptive such that the initial value of tMinHigh is doubled if the timer tMinLow has not expired when the qHigh threshold is crossed. The ALR time-out-threshold policy requires a heuristic setting of the tMinLow and tMinHigh values.

3 Markov Model of the Dual-Threshold Policy

Assuming Poisson arrivals and exponential service times, the ALR dual-threshold policy can be modeled as a state-dependent service-rate, single-server queue where rate transition cannot occur during a service (that is, a service period must complete before the rate can transition). In the case of an Ethernet link, rate transition clearly cannot occur during a packet transmission (a service period); it can only occur after the completion of packet transmission. In this section, we model single and dual-threshold systems with rate transition possible only on service period completion.

We define that packets arrive at a rate λ and are serviced at a low service rate μ_1 or a high service rate μ $(\mu_1 < \mu)$ with respective utilizations as $\rho_1 = \lambda/\mu_1$ and $\rho = \lambda/\mu$, where $\rho < 1$ for stability. In a single-threshold policy, the output buffer occupancy level equaling or exceeding a threshold value k causes the service rate to switch from μ_1 to μ. An output buffer queue length dropping below k causes the service rate to switch from μ to μ_1. In a dual-threshold policy, two buffer occupancy thresholds k_1 and k_2 are defined, where $k_1 < k_2$ (thus, k_1 corresponds to qLow and k_2 to qHigh). An output buffer occupancy level equaling or exceeding threshold value k_2 causes the service rate to switch from μ_1 to μ if the present rate is μ_1. An output buffer occupancy level dropping below threshold value k_1 causes the data rate to switch from μ to μ_1 if the present rate is μ.

Of interest are the packet delay and time spent in the low service rate as a function of the threshold values (k_1 and k_2), the arrival and services rates (λ, μ_1, and μ), and the rate-switching time (T_{switch}). We seek to develop performance models to gain insight into the effects of these parameters on packet delay and energy saved. Energy is saved when the link is in the low service rate (low data rate). We derive expressions for the steady-state probability of n customers (packets) in the system, P_n, and the mean number of customers in the system, L. In the derivations, we use π_n to denote the steady-state probability of state n and, unless explicitly stated, P_n does not equal π_n. The relative time in the low rate is the sum of the steady-state probabilities of states with service rate μ_1.

3.1 Derivation of Steady-State Probabilities

The simplest system is that of a single-threshold queue with a state-dependent service rate, where a service rate transition can occur during a service period. For this system, closed-form expressions are readily available [14].

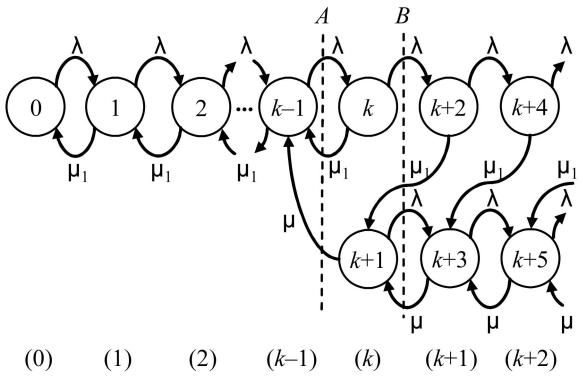

Fig. 7. Single-threshold transition at service completion.

Extending the single-threshold model to two thresholds was first done by Gebhard [10].

To correctly model ALR for Ethernet, rate change can only occur at a service completion (that is, at the completion of sending a packet). Fig. 7 shows the Markov chain for this single-threshold rate change at a service completion system, which is similar to the Markov chain developed by Chong and Zhao [4] for CPU resource scheduling. We solve the Markov chain using direct methods (Chong and Zhao used transforms) and then extend our solution method to solve the Markov chain with two thresholds (Fig. 8). The states $k, k+2, k+4, \ldots k+2n \ \ (n \geq 0)$ model the case where arrivals causing a threshold crossing occur during a service period (and the rate transition occurs first on the completion of the current service). The steady state probabilities of these states are the probability of receiving n arrivals during a service interval. The cumulative probability of receiving an arrival before time t is

$$F(t) = \int_0^t \lambda e^{-\lambda y} dy = 1 - e^{-\lambda t}. \tag{1}$$

Due to the memoryless property of exponentially distributed events, the probability of receiving an arrival before the current service (with rate μ_1) is completed is

$$\int_0^\infty \left(1 - e^{-\lambda t}\right) \mu_1 e^{-\mu_1 t} dt = \frac{\lambda}{\lambda + \mu_1}. \tag{2}$$

Thus, the steady-state probabilities for states $k, k+2, k+4, \ldots$ are related by

$$\pi_k = \left(\frac{\lambda}{\lambda + \mu_1}\right) \pi_{k-1} \tag{3}$$

and, for $n \geq 1$,

$$\pi_{k+2n} = \left(\frac{\lambda}{\lambda + \mu_1}\right) \pi_{k+2n-2}, \tag{4}$$

giving that for $n \geq 1$,

$$\pi_{k+2n} = \left(\frac{\lambda}{\lambda + \mu_1}\right)^{n+1} \pi_{k-1}. \tag{5}$$

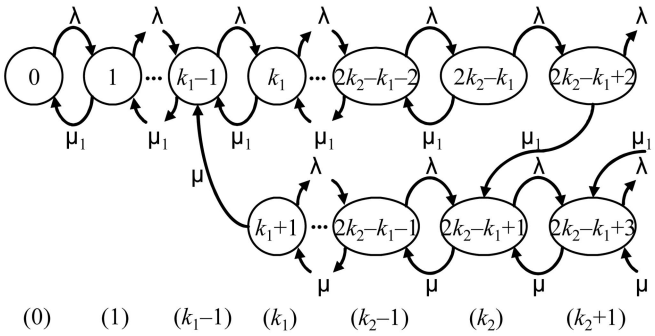

Fig. 8. Dual-threshold transition at service completion.

By partitioning the Markov chain into two subsets with no shared states and writing the balance equations for the state transitions between the subsets, we are able to derive the steady state probabilities for the number of customers in the system. Up to state $k-1$, the Markov chain is similar to an M/M/1 queue and, therefore,

$$P_{k-1} = \pi_{k-1} = P_0 \rho_1^{k-1}. \tag{6}$$

Partitioning the chain on cut (A) yields

$$\lambda \pi_{k-1} = \mu_1 \pi_k + \mu \pi_{k+1}. \tag{7}$$

Partitioning the chain on cut (B) yields

$$\lambda \pi_k + \lambda \pi_{k+1} = \mu_1 \pi_{k+2} + \mu \pi_{k+3}. \tag{8}$$

In Fig. 7, we see that

$$P_k = \pi_k + \pi_{k+1}. \tag{9}$$

By substituting from (5) and (6), (9) can be written as

$$\pi_{k+1} = P_k - P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right). \tag{10}$$

Similarly, we can write

$$\pi_{k+3} = P_{k+1} - P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right)^2. \tag{11}$$

By substituting from (5) and (10) in (7), we get

$$\lambda P_{k-1} = \mu_1 P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right) + \mu\left(P_k - P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right)\right). \tag{12}$$

By substituting from (5), (10), and (11) in (8), we get

$$\lambda P_k = \mu_1 P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right)^2 + \mu\left(P_{k+1} - P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right)^2\right). \tag{13}$$

Finally, from (12) and (13), we can generate the following general equation for steady-state probabilities, where $n \geq k$:

$$\lambda P_{n-1} = \mu_1 P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right)^{n-k+1} + \mu\left(P_n - P_{k-1}\left(\frac{\lambda}{\lambda + \mu_1}\right)^{n-k+1}\right). \tag{14}$$

Equation (14) can be refined to yield the following recursive equation for the steady state probability of having n customers in the system, where $n \geq k$,

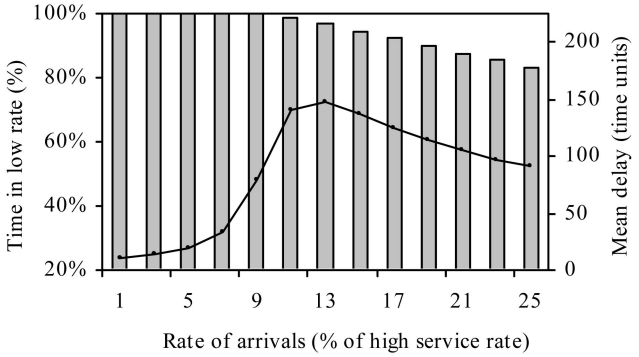

Fig. 9. Numerical results for the Markov model of the dual-threshold policy.

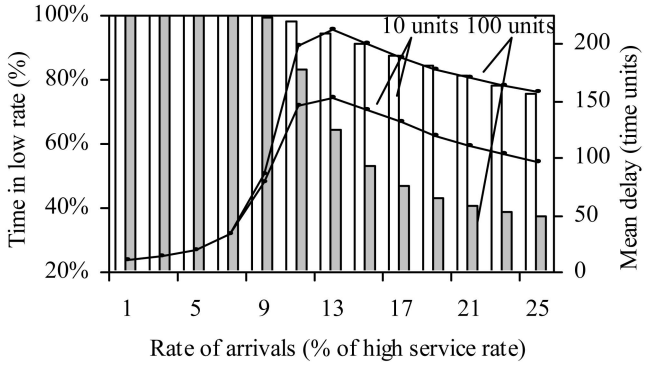

Fig. 10. Simulation results with nonzero switching time of the dual-threshold policy.

$$P_n = \rho P_{n-1} + \left(1 - \frac{\rho}{\rho_1}\right) P_{k-1} \left(\frac{\rho_1}{1+\rho_1}\right)^{n-k+1}. \quad (15)$$

Equation (15) can be iteratively solved and the final closed-form equations for the steady state probabilities and P_0 are

$$P_n = \begin{cases} P_0 \rho_1^n & (0 \le n < k), \\ \frac{P_0 \rho_1^{k-1}}{\rho+\rho/\rho_1-1} \left(\rho^{n-k+2} - \left(1 - \frac{\rho}{\rho_1}\right) \left(\frac{\rho_1}{1+\rho_1}\right)^{n-k+1} \right) & (n \ge k), \end{cases} \quad (16)$$

$$P_0 = \left(\frac{1-\rho_1^k}{1-\rho_1} + \frac{\rho_1^k}{1-\rho} \right)^{-1}. \quad (17)$$

The single threshold with rate transition at service completion model extended to a dual-threshold model is shown in Fig. 8. Although more complex than the Markov chain for the single-threshold case, similar techniques can be used to derive the steady state probabilities. The steady state probabilities are given by

$$P_n = \begin{cases} P_0 \rho_1^n & (0 \le n < k_1) \\ \frac{P_0 \rho_1^{k_1-1}(1-1/\rho_1)}{1-1/\rho_1^{k_2-k_1+2}} \left(\frac{1-1/\rho_1^{k_2-n+1}}{1-1/\rho_1} + \frac{\rho\left(1-\rho^{n-k_1+1}\right)}{1-\rho} \right) & (k_1 \le n < k_2) \\ \frac{P_0 \rho_1^{k_1-1}}{1-1/\rho_1^{k_2-k_1+2}} & \\ \left(\rho^{n-k_2+1} \left(\frac{(1-1/\rho_1)\left(1-\rho^{k_2-k_1}\right)}{1-\rho} - \frac{1-1/\rho_1^2}{1-\rho-\rho/\rho_1} \right) \right. & (n \ge k_2), \\ \left. + \frac{\left(1-1/\rho_1^2\right)(1-\rho/\rho_1)}{1-\rho-\rho/\rho_1} \left(\frac{\rho_1}{1+\rho_1}\right)^{n-k_2+1} \right) & \end{cases} \quad (18)$$

and P_0 is given by

$$P_0 = \left(\frac{1-\rho_1^{k_1}}{1-\rho_1} + \frac{1}{1-\rho_1^{k_2-k_1+2}} \left(\frac{\rho_1^{k_2} - \rho_1^{k_1}}{\rho_1 - 1} + \frac{\rho_1^{k_2}\left((k_2-k_1)(\rho_1-\rho) + \rho_1^2 - 1\right)}{\rho - 1} \right) \right)^{-1}. \quad (19)$$

The increase in mean queue length caused by switching service rates at the end of a service completion can be shown to be bounded by λ/μ_1, which is the mean number of new arrivals that can be received during one service interval at service rate μ_1.

3.2 Numerical Results from the Markov Models

The fraction of time in the low rate and the mean packet delay are measured using the dual-threshold rate transition at service completion Markov model in Fig. 8. The low service rate μ_1 was set to 0.1 and the high service rate μ was set to 1.0 (this models Ethernet data rates, which increase in multiples of 10). The threshold values k_1 and k_2 were set to 15 and 30 customers, respectively (noting that NIC buffers are generally about 32 Kbytes [2], which corresponds to approximately 30 large packets). The rate of arrivals λ was varied from 1 percent to 25 percent of the high rate μ. The numerical results for this case are shown in Fig. 9, where the bars show the percentage of time in the low rate and the line shows the mean packet delay. The time units for measuring delay are the units of the mean service interval at the high service rate μ. It is observed that, initially, the time in the low rate is 100 percent and the mean queue length (and, thus, also the mean response time) increases with increasing λ. In this region, the queue length is insufficient to trigger a change to the high service rate. However, when λ exceeds μ_1, the queue length exceeds the upper threshold k_2, triggering a service rate change to μ. As the difference between λ and μ_1 increases, the proportion of packets serviced at the higher rate increases and, consequently, the mean response time decreases.

A simulation model (which was first validated by exactly reproducing the numerical results in Fig. 9) was built using the CSIM19 discrete event simulation library [33] and was used to repeat the previous experiment with nonzero rate switching transition times (T_{switch}) of 10 and 100 mean service intervals at the high service rate μ. The simulation results are shown in Figs. 10 and 11. The time in the low rate and the mean response time are shown in Fig. 10 and the number of service rate transitions per 1,000 time units for switching times of 0, 10, and 100 time units as the rate of arrivals is increased is shown in Fig. 11. In Figs. 9 and 10, we observe that, for up to 9 percent offered load (relative to the high service rate), the percentage of time in the low rate is approximately 100 percent and constant. In this region, the system behaves as an M/M/1 queue at 90 percent offered load with a mean queue length of nine customers. With an increase in the rate of arrivals, the fraction of time in the low rate decreases, with the biggest decrease shown for the largest switching time.

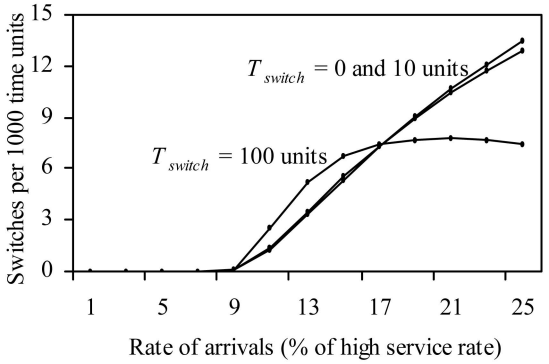

Fig. 11. Rate switches per 1,000 time units for the dual-threshold policy (from a simulation model).

In Fig. 11, the number of rate switches is shown for various nonzero switching times. For switching times of 0 and 10 time units, the number of switches increases with the increasing arrival rate λ. However, for 100 time units of switching time, the number of rate switches increases initially and then starts to decrease. This is due to the greater number of arrivals queued during the longer rate switching time, which reduces the time spent servicing arrivals at rate μ_1 as the upper threshold k_2 is exceeded more quickly than in the case of 0 or 10 time unit switching time. As λ increases, a greater number of packets is queued during each rate switch and the time to drain the queue at rate μ increases, causing less rate switching activity. Eventually, the time spent servicing arrivals at low service rate μ_1 becomes insignificant with increasing λ. This is because, by the time the service rate change is complete, the queue length is again greater than threshold k_2. At a rate of arrivals of 25 percent of the high service rate and a switching time of 100 time units, the time in the low rate is less than 40 percent (this can be seen in Fig. 10) and the majority of time is spent in rate switching. The dual-threshold policy for data rates of 100 Mbps and 1 Gbps at 15 percent offered load would switch the link data rate more than 400 times per second. For a T_{switch} of 1 ms, this is more than 40 percent of the time spent on switching data rates.

3.3 Data Rate Oscillations in ALR Policies

The number of rate switches during a unit time period for the dual-threshold policy (N_{dual}) can be computed using the Markov model in Fig. 8. The number of rate switches during a unit time period is given by the number of state transitions between states with differing rates of service. The number of state transitions during a unit time period out of a given state is given by the steady state probability of being in that state multiplied by the rate of leaving that state. The number of rate transitions in a unit time period is thus

$$N_{dual} = \mu P_{k_1+1} + \sum_{n=1}^{\infty} \mu_1 P_{2k_2-k_1+2n}. \tag{20}$$

The service rate changes from μ to μ_1 when transitioning from state k_1+1 to k_1-1 (this is the first term in (20)) and changes from μ_1 to μ when transitioning from states $2k_2-k_1+2n$ to $2k_2-k_1+2n-1$ $(n \geq 1)$ (this is the second term in (20)).

The utilization-threshold policy cannot be modeled as a Markov chain due to the fixed tUtil time period. We model the oscillation behavior of the utilization-threshold policy by solving for the mean passage time between queue thresholds in an M/M/1 queue with two service rates, μ_1 and μ. The mean time of an oscillation is modeled as the time between successive rate transitions from μ_1 to μ or from μ to μ_1 for a given arrival rate, λ. The mean time between successive rate transitions is the sum of three separate time periods. The three time periods are the following:

1. The mean passage time $(T_{passLow})$ from the low queue threshold (k_1) to the high queue threshold (k_2) at low service rate (μ_1). When the queue length reaches k_2 packets, additional packets may arrive during the current service time (that is, during the packet transmission time). Thus, $T_{passLow}$ is computed from queue length k_1 to queue length k_2 plus the mean number of arrivals during a service time. The mean number of arrivals during a service is computed using (5). The value of $T_{passLow}$ is computed iteratively (see, for example, [25]).
2. The mean passage time $(T_{passHigh})$ from the high queue threshold (k_2) to the low queue threshold (k_1) at high service rate (μ). Similarly to $T_{passLow}$, the value of $T_{passHigh}$ is computed iteratively.
3. The mean time (T_{high}) in service rate μ once the queue length has dropped to k_1. In order for the data rate to change from μ to μ_1, the queue length must be less than the low threshold k_1 and the value of uBytes must be less than uThresh at the end of a tUtil time period. Thus, (T_{high}) is determined by the probability of uBytes being less than uThresh (p) and the probability of the queue length being less than the threshold k_1 at the end of a tUtil time period. The time period T_{high} is distributed (approximately) geometrically:

$$T_{high} = \frac{tUtil}{p \sum_{n=0}^{k_1-1} \pi_n}, \tag{21}$$

where π_n is the steady state probability of having n customers in the system for an M/M/1 queue with a rate of service μ. This is an approximation since it assumes that the steady state M/M/1 behavior holds during the time the queue length is less than k_1 at service rate μ.

Each oscillation has two rates switches and, thus, the number of oscillations during a unit time period for the utilization-threshold policy (N_{util}) is

$$N_{util} = \frac{2}{\left(T_{passLow} + T_{passHigh} + T_{high}\right)}. \tag{22}$$

The rate oscillation for the time-out-threshold policy with a fixed (nonadaptive) initial tMinHigh value can be modeled in a similar fashion. The time in the low rate (μ_1) is the mean passage time $T_{passLow}$ and is computed as described above. The time in the high rate (μ) is T_{high} and is simply the heuristically assigned value of tMinHigh (given that tMinHigh is larger than the time for the queue to transition from k_2 to k_1, which we assume is the case). Thus, the number of rate switches during a unit time period for the time-out-threshold policy $(N_{timeout})$ is

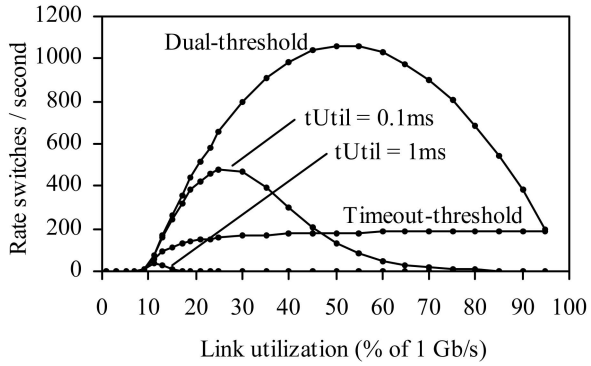

Fig. 12. Rate oscillations for ALR policies.

$$N_{timeout} = \frac{2}{T_{passLow} + T_{high}}. \tag{23}$$

The number of rate switches per second when using the ALR policies was evaluated using both the analytical expressions ((20), (22), and (23)) and the simulation models. We assumed Poisson arrivals with a mean packet size of 1,500 bytes and link data rates of 100 Mbps and 1 Gbps, which results in service rates $\mu_1 = 8,333.33$ and $\mu = 83,333.33$. The arrival rate λ was varied as a percentage of the service rate μ. The queue length threshold k_1 was set to 1 and k_2 was set to 30. For all cases, it was assumed that the switching time was negligible (that is, $T_{switch} = 0$). The tUtil time period was set to 0.1 and 1 ms for the utilization-threshold policy. For the time-out-threshold policy, tMinHigh and tMinLow were both set to 10 ms.

Fig. 12 shows the results from (20) for the dual-threshold policy, (22) for the utilization-threshold policy, and (23) for the nonadaptive time-out-threshold policy. In almost all cases, the values from the simulation models and the equations are within 3 percent and, thus, the simulation model results are not shown in Fig. 12. The only case where the simulation model and equations do not match closely is for the time-out-threshold policy when the link utilization is greater than 80 percent. For this case, the mean queue length is greater than the threshold k_1; therefore, the time spent in the high rate is greater than tMinHigh resulting in fewer rate switches than that given by (23). These results show that, for the dual-threshold policy, once the rate of arrivals λ is greater than the low service rate μ_1, a rate switch to rate μ_1 will inevitably result in a switch back to the high service rate μ in a short period of time. Since, in practice, the rate switching time is nonzero, this results in additional packet delay. Ideally, once λ exceeds μ_1, no rate switches to μ_1 should take place. The utilization-threshold policy is the most effective policy for reducing data rate oscillations. Increasing the tUtil time period reduces oscillations. At a link utilization of 20 percent, a $\mathrm{tUtil} = 0.1$ ms results in about 400 rate switches per second, whereas a $\mathrm{tUtil} = 1$ ms results in no rate switches. Determining the optimal tUtil time period is described in the next section. It can be seen that the nonadaptive time-out-threshold policy performs between the dual-threshold and utilization-threshold policies.

4 Sampling Time for the Utilization-Threshold Policy

The minimum value of tUtil needed to achieve a given margin of error and level of confidence can be derived for an arrival process with known (and finite) first and second moments. From the central limit theorem, the probability of a population mean, $1/\lambda$ (where λ is the mean rate of arrivals), being within a confidence interval is

$$\Pr\left(\frac{T}{n} - \frac{a}{\lambda} < \frac{1}{\mu} < \frac{T}{n} + \frac{a}{\lambda}\right) = p, \tag{24}$$

where

$$\frac{a}{\lambda} = \frac{zs}{\sqrt{n}} \tag{25}$$

and T is the time period during which observations were recorded, s is the standard deviation of the sample, a is the desired accuracy $(0 < a < 1)$, z is the normal variate for the desired confidence interval ($z = 1.96$ for 95 percent confidence), and n is the number of arrivals. The number of arrivals required for a margin of error of a/λ is then

$$n = \frac{z^2 s^2 \lambda^2}{a^2}. \tag{26}$$

For exponentially distributed interarrival times, $s^2 = 1/\lambda^2$ and, thus, (26) can be simplified to

$$n = \frac{z^2}{a^2}. \tag{27}$$

Since n is known, the optimal T (that is, the optimal tUtil value) can now be calculated for a given rate of arrivals as

$$T = \frac{z^2}{a^2 \lambda}. \tag{28}$$

The number of samples (n) will correspond to the value of tUtil in seconds. For a link date rate of 1 Gbps, fixed-length 1,500 byte packets, and 5 percent link utilization, the mean rate of arrivals (λ) is 4,166.67 packet arrivals per second. With $a = 0.1$ and $z = 1.96$, we get $\mathrm{tUtil} = 92.2$ ms. In 92.2 ms at 5 percent utilization of 1 Gbps, uThresh is 576,250 bytes. At utilizations greater than 50 percent, packet delay increases rapidly. Therefore, we seek to operate the link at the low data rate for utilizations less than 50 percent measured at the low data rate, which is 5 percent utilization at the high data rate.

5 Simulation Modeling of ALR

In this section, we study the behavior and performance of the ALR policies for non-Poisson traffic using the simulation models described and validated previously. The metrics of interest are the mean response time (packet delay), the time in the low rate (energy savings), and the time spent switching between rates (oscillation time). The parameter uThresh is a link utilization percentage during time period tUtil and is defined to be the number of bytes that can be transmitted at 5 percent link utilization in time tUtil at the high data rate. The link will be prevented from switching to the low rate if the number of bytes transmitted during a tUtil time period exceeds the value of uThresh.

TABLE 1
Characteristics of Actual (Traced) Traffic

Trace	Burst Len	CoV	Hurst	Util (1)
USF #1	17.5 KB	1.76	0.66	4.20 %
USF #2	14.6	2.18	0.76	2.63
USF #3	24.1	13.95	0.82	0.03
PSU #1	26.7	2.21	0.73	0.13
PSU #2	642.8	6.07	0.91	1.01
PSU #3	24.4	3.54	0.91	1.03

(1) Computed at 100 Mbps (1,518 byte packets assumed for PSU).

5.1 Traffic Models

The ALR policies need to be effective for 1 and 10 Gbps Ethernet links. Characterizations of existing 100 Mbps desktop-to-LAN-switch Ethernet links show that traffic is very bursty. Table 1 shows the traffic characteristics of six traffic traces, each of duration 30 minutes or greater, taken from Ethernet links at the University of South Florida (USF) and Portland State University (PSU) [19]. In Table 1, a "burst" is defined as one or more consecutive 10 ms time periods with 5 percent or greater utilization. It is reasonable to believe that traffic on 1 and 10 Gbps Ethernet links will likely have very similar characteristics, where the majority of traffic volume is in bursts and background "network chatter" (for example, ARP broadcasts) is negligible. File transfer applications such as P2P file sharing are one common source of bursty traffic.

In [9], Ethernet traffic is modeled as bursty with Cauchy distributed burst lengths and exponentially distributed idle interburst times. Due to the lack of a mean, the Cauchy distribution can be difficult to use in practice. In [7], it is shown that traffic bursts follow a bounded Pareto distribution, which follows from the distribution of sizes of files stored in Web servers. The probability distribution for the bounded Pareto is

$$f(x) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} x^{-(\alpha+1)}, \tag{29}$$

where k and p are the lower and upper bounds on $f(x)$, and α is the Pareto index.

We implemented a synthetic traffic generator to generate a synthetic traffic trace (a text file containing timestamp and packet length pairs). Burst size is bounded Pareto distributed and interburst idle time is exponentially distributed. Packet length is fixed or empirically distributed. Fig. 13 shows a time plot of packet counts for actual and synthetic 100 Mbps traffic where the actual traffic is from the USF #1 trace. Two time scales (with 10 and 100 ms bins) visually show that the actual and synthetic traffic are bursty across multiple time scales, as would be expected for traffic with a Hurst parameter greater than 0.5. The parameter values used for the synthetic traffic were $k = 1,518$ bytes, $p = 1$ Gbyte, $\alpha = 1.5$, burst intensity of 80 percent, target utilization of 4 percent, and an empirical packet size distribution based upon the actual packet size distribution of the USF #1 trace. The CDF of the packet counts for 10 and 100 ms bins for the actual and synthetic traces are shown in Fig. 14. It can be observed that the CDF for the actual and synthetic traces are very similar. Table 2 shows the summary statistics of the two traces, further demonstrating the ability of the traffic generator to generate synthetic traffic with realistic values including utilization, mean interpacket time, and the Hurst parameter, all within about 5 percent of actual. Note the large Hurst parameter value that indicates a high degree of self-similarity in the traffic.

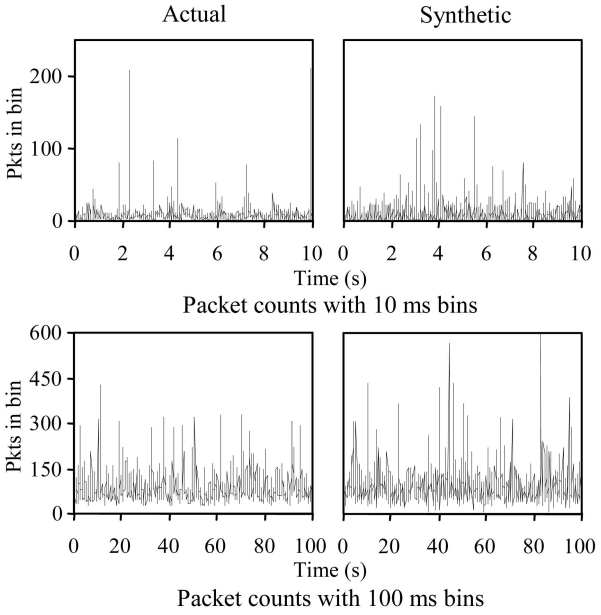

Fig. 13. Packet counts for different time scales showing burstiness.

5.2 Simulation Experiments

The dual-threshold, utilization-threshold, and time-out-threshold policies were implemented in simulation models of an Ethernet NIC and a 16-port output queued switch. The Ethernet NIC model is the same as described and validated in Section 3. The switch model was developed and validated in [38]. For this study, ALR has been added to the output buffers.

Simulation experiments were designed to evaluate the dual-threshold, utilization-threshold, and time-out-threshold ALR policies. For the experiments described below, unless stated otherwise, $\mathrm{qLow} = 0$ Kbyte, $\mathrm{qHigh} = 32$ Kbyte, the low and high data rates were 100 Mbps and 1 Gbps, and $T_{switch} = 1$ ms. It was assumed that all ALR data rate switch requests are successfully ACKed. All experiments were run for at least 10 million packet arrivals. Unless otherwise noted, the values of tUtil were 0.1, 1, 10, and 100 ms. For the adaptive time-out-threshold policy, the tMinLow values

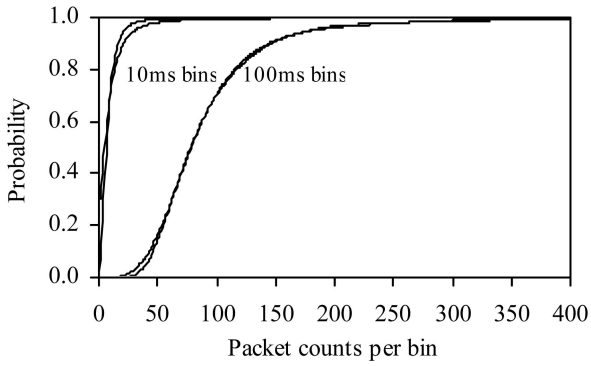

Fig. 14. CDF plots for the packet counts of 10 and 100 ms bins.

TABLE 2
Actual versus Synthetic Traffic

Characteristic	Actual	Synthetic
Mean inter-packet time (ms)	1.10	1.06
CoV of inter-packet times	1.76	3.81
Mean packet size (bytes)	577	526
CoV of packet size	1.16	1.15
Hurst parameter of packet counts	0.66	0.64
Utilization (percentage of 100 Mbps)	4.20	3.97

used were 10, 50, and 100 ms (which is 10, 50, and 100 times the link rate switching time, respectively). The initial tMinHigh value was set to 10 ms in all cases.

Two scenarios for future high-bandwidth Ethernet traffic were considered. Multiplayer gaming, virtual reality environments, and collaborative file sharing applications all generate relatively small bursts of data with millisecond-scale interburst times. This scenario was evaluated in *Bursty traffic experiment #1*. A corporate or scientific network where gigabyte-scale data sets and backups are transferred several times per day (that is, with many minutes or hours between bursts) was evaluated in *Bursty traffic experiment #2*. Network "background chatter" between bursts was considered to be negligible compared to the data rate. The experiments were:

Smooth traffic experiment. The effect of varying tUtil for Poisson (smooth) traffic was studied in this experiment. A fixed-length packet size of 1,500 bytes was used. The traffic utilization ranged from 1 percent to 95 percent.

Single burst experiment. The transient effects of rate switching were studied in this experiment. Specifically, a single burst comprised of Poisson traffic at 80 percent utilization and 0.4 seconds in duration was studied.

Bursty traffic experiment #1. The effect of bursty traffic with small bursts was studied in this experiment. The traffic generator was used to generate synthetic traffic traces with a mean burst size of 8.4 Kbytes and target utilization ranging from 1 percent to 25 percent. The parameter values were $k = 1,518$ bytes, $p = 2.5$ Gbytes, $\alpha = 1.5$, and burst intensity of 80 percent. The resulting Hurst parameter and CoV were, respectively, 0.80 and 3.35 for 1 percent utilization and 0.81 and 2.33 for 25 percent utilization.

Bursty traffic experiment #2. The effect of bursty traffic with large bursts was studied in this experiment. The traffic generator was used to generate synthetic traffic traces with a mean burst size of 867 Mbytes and interburst times ranging from 10 seconds to 100,000 seconds. The parameter values were $k = 250$ Mbytes, $p = 10$ Gbytes, $\alpha = 1.1$, and burst intensity of 80 percent. The tUtil values evaluated were 1, 10, and 100 ms.

Switch experiment. This experiment studied measurable energy savings in a realistic LAN switch configuration. In [16], the power consumption of a Cisco Catalyst 2970 switch with no Ethernet links attached was measured to be 46 W. Each Ethernet link operating at 100 Mbps or 1 Gbps added an additional (measured) 0.3 W or 1.8 W, respectively, to the switch power consumption. All power measurements were made at the wall socket. We modeled a switch with 16 full-duplex Ethernet ports where input traffic streams were generated with the parameters used for *Bursty traffic experiment #1*. Utilization (switch offered load) was increased from 1 percent to 15 percent of the 1 Gbps high data rate. All packets in a burst had the same destination port in the switch. The metrics of interest were power consumption (W) and the mean switch delay for packets (time difference between entry and exit for a packet). The values of tUtil studied were 10 and 100 ms.

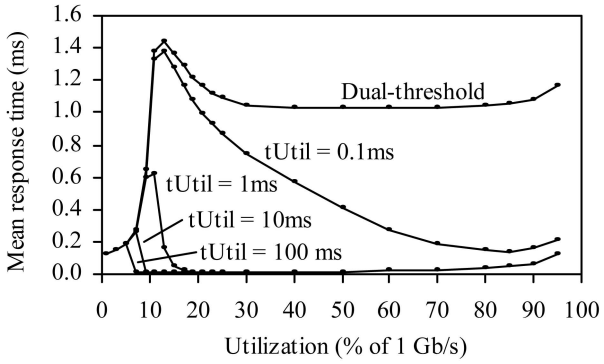

Fig. 15. Results from the smooth traffic experiment.

5.3 Experiment Results and Discussion

The results from the *Smooth traffic experiment* are shown in Fig. 15. It can be observed that, for $\mathrm{tUtil} = 100$ ms, the mean response time was reduced when utilization exceeded 5 percent (at the high data rate) due to the utilization-threshold policy detecting that the link utilization was greater than 5 percent and maintaining a constant 1 Gbps link data rate. It can be seen that, as the tUtil value decreased to 0.1 ms, the link utilization level at which the mean response time reaches its peak increases. This was due to the length of tUtil being insufficient to receive the number of bytes that denotes link utilizations greater than 5 percent during all tUtil time periods. Therefore, at the end of time periods where $\mathrm{uBytes} < \mathrm{uThresh}$, the link rate was set to 100 Mbps and this resulted in a greater response time, both due to the lower service rate and rate switching time. In this experiment, we see that the analysis in Section 4 is validated by experimental results.

The results for the *Single burst experiment* are shown in Fig. 16 (only for $\mathrm{tUtil} = 0.1$ ms) and Table 3. A spike in the response time denotes a rate switch. The initial rate switch occurred at $\mathrm{time} = 0.1$ second (which is the burst start time). For $\mathrm{tUtil} = 0.1$ ms, it is observed that five rate switch oscillations occurred. These oscillations increased the mean response time. The mean response time was 0.19 ms with the 90th percentile at 0.73 ms and the 99th percentile at 1.88 ms. In Table 3, the number of rate switches, the mean response time,

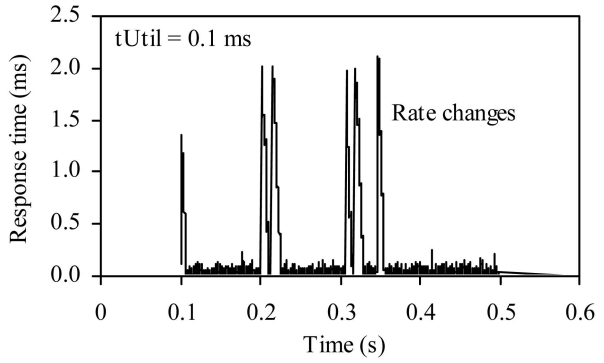

Fig. 16. Results from the single burst experiment.

TABLE 3
Results from the Single Burst Experiment

Characteristic	tUtil value (ms)			
	0.1	1	10	100
Rate changes	12	2	2	2
Mean response time (ms)	0.19	0.05	0.05	0.05
Post burst lag (ms)	0.19	1.19	12.2	102

and the postburst lag for varying tUtil values are shown. The postburst lag time gives the time difference between the end of a burst and the final rate switch to 100 Mbps.

The results for *Bursty traffic experiment #1* for the utilization-threshold and dual-threshold policies are shown in Figs. 17 and 18. As tUtil and link utilization was increased, the mean response time and time in the low rate decreased. From the previous experiment, we know that the postburst lag increases with an increasing tUtil value. With greater postburst lag time, the possibility that the next burst will begin before the link data rate switches to 100 Mbps increases and, therefore, the time in the low rate decreases. With increasing utilization, the interburst time decreases and, therefore, the possibility that, for a given tUtil value, the next burst will start before the data rate changes to 100 Mbps increases. Consequently, fewer packets are serviced at the lower data rate and the mean delay is decreased. With the dual-threshold policy, we see that rate switch oscillations caused the mean response time to increase as the utilization increased. The results for the time-out-threshold and dual-threshold policies are shown in Figs. 19 and 20. The results are roughly comparable to the utilization-threshold policy. The number of low-rate time periods greater than tMinLow decreases as the tMinLow value increases and the resulting greater tMinHigh value reduced the time in the low rate.

For *Bursty traffic experiment #2*, we found that, for the utilization-threshold policy, the data rate switched from 100 Mbps to 1 Gbps at the beginning of a burst and returned to the 100 Mbps data rate at the end of the burst. With the utilization-threshold policy, there were no rate switches during the bursts. The postburst lag time was insignificant compared to the interburst times of 10 to 100,000 seconds and, thus, virtually all nonburst time was spent in the low (and energy saving) 100 Mbps data rate. With the dual-threshold policy, we found that the link data rate oscillated between 100 Mbps and 1 Gbps during the bursts, leading to a mean response time of more than 1 ms. For the adaptive time-out-threshold policy, we found that the average mean response time was approximately twice that of the utilization-threshold policy (0.025 ms versus 0.013 ms) due to rate switches during bursts and that the time in the 100 Mbps data rate was approximately the same as for the utilization-threshold policy.

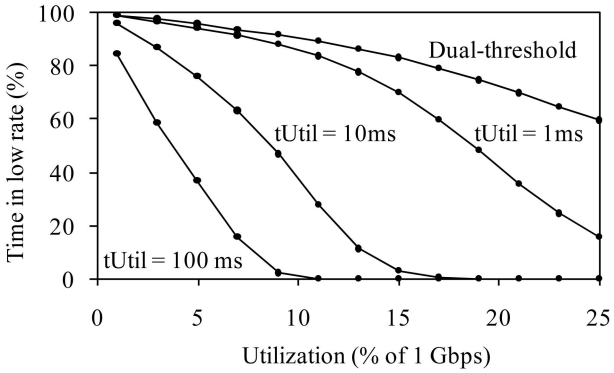

Fig. 18. Time in low rate from bursty traffic experiment #1 for the utilization-threshold policy.

The results for the *Switch experiment* are shown in Fig. 21. The switch power consumption is given for $\mathrm{tUtil} = 10$ ms and 100 ms (shown as bars), with the dashed line indicating switch power consumption without ALR. The solid lines show the mean response time for $\mathrm{tUtil} = 10$ ms and 100 ms and with ALR disabled (that is, a constant 1 Gbps data rate). We see that, with an average utilization of 5 percent, a power savings of about 15 W (20 percent) is possible for $\mathrm{tUtil} = 10$ ms. The increase in mean response time when ALR is enabled for both values of tUtil was less than 0.5 ms, which can be considered negligible for most applications.

6 Potential Energy Savings from ALR

Estimating future energy savings from the widespread deployment of ALR is difficult due to the large number of variables that need to be considered. We first present the savings calculation for commercial-sector desktop PCs, which are the largest single determinant of ALR savings. This is followed by results for a wider selection of residential and commercial products. The estimates presented here

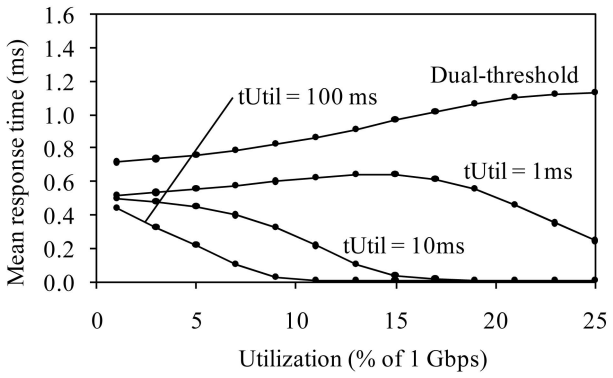

Fig. 17. Response time from bursty traffic experiment #1 for the utilization-threshold policy.

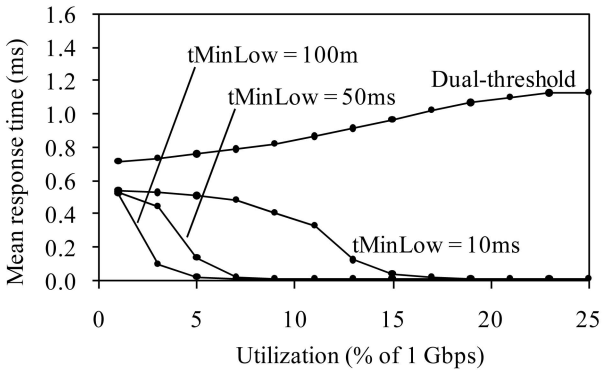

Fig. 19. Response time from bursty traffic experiment #1 for the adaptive time-out-threshold policy.

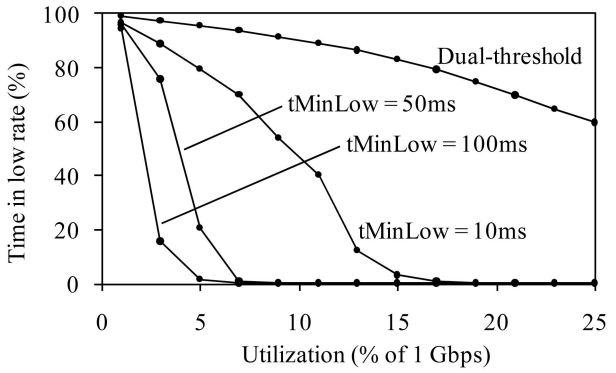

Fig. 20. Time in low rate from bursty traffic experiment #1 for the adaptive time-out-threshold policy.

reflect 100 percent of 1 Gbps and 10 Gbps Ethernet links using ALR. Although ALR will not reach 100 percent of links, these figures do not include savings beyond the US and no savings are estimated for reduced building space conditioning. For a conservative estimate, we make the following assumptions:

- The typical workday is 8 hours plus a 1 hour lunch.
- Consistent with field data, 2/3 of PCs are left on at nights and during weekends [30], [37].
- A full 1 Gbps data rate is required only 1 hour per workday (for the rest of the time, operation at 100 Mbps is possible with no perceivable difference to the user).
- The power difference between 100 Mbps and 1 Gbps is 2 W for a desktop PC to switch link (with 1 W of savings in the desktop PC and 1 W in the switch).
- There are 72 million Ethernet-connected commercial desktop PCs.

With widespread use of ALR, about $70 million in energy savings per year in the US alone for the commercial sector is possible. Table 4 shows the assumed and calculated values for this savings estimate; similar calculations are done for other product types.

The full estimate covers a variety of product types. Residential products that contribute more than 5 percent of the 1 Gbps total savings are desktop PCs, cable and DSL modems, routers, and wireless access points. Commercial products over 5 percent are desktop PCs, printers, and LAN switches. Stock values are adapted from [27] and [32]. The

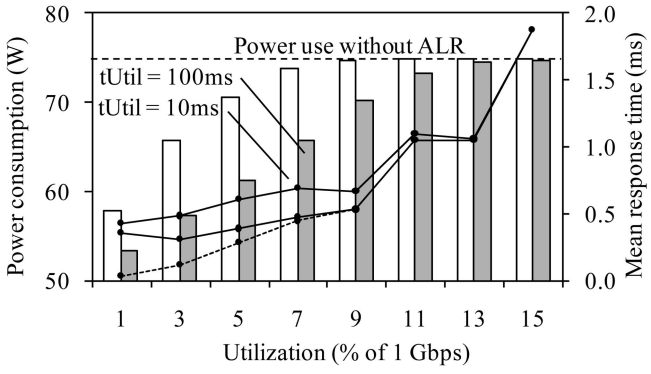

Fig. 21. Results from the switch experiment.

TABLE 4
Assumptions and Calculations for Commercial Desktop PC Links

Parameter	Assumed	Calculated
Total stock (desktop PCs only)	80 million	
Ethernet connected fraction	90 %	
Ethernet connected PCs		72 million
Number of NICs/product	1	
Work-time potential low-rate time (lunch and one hour of high-rate time)	8 hrs/day	
Night and weekend full-on time	67 %	
Weekly potential ALR low-rate time		About 70 % 6100 hours/yr
Annual savings per ALR PC link		12.2 kWh/yr
Annual savings per ALR PC link (at $0.08/kWh)		$ 0.98
ALR link penetration	100 %	
Savings for 72 million ALR PC links		$ 70 million/yr

assumptions for the percentage of NICs with a cable connected and the percentage of potential ALR low-speed time are estimates. The total estimate for savings from ALR is $300 million/year, with nearly half from residential products and about 14 percent from 10 Gbps ALR in data centers.

The potential for energy savings is so great that the revised EPA Energy Star Program Requirements for Computers version 4.0 [36] states that "All computers shall reduce their network link speeds during times of low data traffic levels in accordance with any industry standards that provides for quick transitions among link rates." An earlier version (Draft 2) further noted that "With such capabilities in place, reduced link rates are expected to be heavily used on Tier 2 computers, and have the potential for significant savings." These statements from the EPA are a direct result of this work on ALR, as first described in [16] and presented to the IEEE in July 2005 [26].

7 Related Work

Methods to reduce the power consumption of computing and communication devices have drawn significant research interest during the past few years. Predictive power management was first proposed in [34]. Dynamic power management has been investigated for disk drives [13], processors [12], and other components. Dynamic power management can be modeled as a finite set of states where finding the optimal power management policy subject to constraints can be formulated as a stochastic optimization problem [1]. In [1], stationary workloads were assumed. In [6], nonstationary workloads were considered by the use of sliding windows of workload history. In [29], the workload is modeled as a Markov-modulated Poisson process and the power management policy that best matches the current workload is selected from a set of precomputed policies. However, these methods require significant computation to derive the optimal policy and to estimate the current workload, which might not be feasible in all cases.

Within the context of mobile devices and wireless networks, considerable research in reducing power consumption has been conducted in order to increase battery lifetime. The lack of legacy protocols and products in the wireless area has enabled novel "clean-slate" solutions—new transport layer protocols, new routing protocols, and new MAC protocols, all designed to permit network

interfaces to power off when not transmitting or receiving data. Wireless networks are either self-contained or connected to the wider Internet through a gateway. The need for interoperability with existing protocols and products does not exist in these new networks and this limits the applicability of these solutions to existing wired networks.

For dynamic power management, solutions for wired networks interoperability with legacy protocols and devices is essential. Widespread adoption of new solutions cannot be expected otherwise. Another factor to consider is that wireless networks mostly operate at significantly lower data rates than wired networks. Correspondingly, the need for buffer capacity can be lower and acceptable latencies can be higher. Reducing the power consumption of wired network links and network devices was first considered in [18] and [5]. In [18], it was shown that the energy efficiency of the wired Internet is less than that of a typical 802.11 wireless LAN. It was argued that, since wireless is a broadcast media, this indicates that the Internet is highly inefficient energy-wise. In [19], it was proposed to power down network interfaces in LAN switches during packet interarrival times. The next packet arrival time is predicted and, if the time interval is greater than a predetermined value, the interface is powered down. Reductions in power consumption of more than 50 percent are shown; however, the effect on packet delay is not discussed.

ADSL2 and ADSL2+ (ITU-T Recommendations G.992.3 [22] and G.992.5 [23]) support multiple link rates and power states [11]. ADSL2+ is the recommended technology for home broadband access in the EU. Although VDSL2 [24], the next generation broadband access technology, has less power management than ADSL2+, the EU Stand-by Initiative has the goal to "trigger action on energy efficiency within standardization" for VDSL2 with the goal of having VDSL2 include the same power management capabilities as ADLS2+ [8].

8 Summary and Future Work

Adaptive Link Rate (ALR) is a method to reduce the energy use of Ethernet links by adapting the link data rate to link utilization. ALR consists of a mechanism and policy. The utilization of Ethernet links is, on average, extremely low. This suggests that there is ample opportunity for energy savings by operating Ethernet links at a low data rate for most of the time with no perceivable performance impact to the user. With an average link utilization of 5 percent or less (measured at the high data rate), it is possible to operate at 100 Mbps for 80 percent or more of the time at an added delay of less than 0.5 ms (for example, see Figs. 17 and 18 for $\mathrm{tUtil} = 10\ \mathrm{ms}$). The estimated energy savings in the US from using ALR in commercial desktop PCs is $70 million per year. When commercial, residential, and data center usage of ALR is included, the estimated savings is over $300 million per year.

We developed a Markov model for a state-dependent service rate, single server queue with dual thresholds for service rate transitions where service rate transition can only occur at the completion of a service period (this correctly models the behavior of a packet switched network). Rate transition at service completion increases the mean number of arrivals in the system and the increase is bounded by λ/μ_1. For smooth traffic (such as with Poisson arrivals), the dual threshold policy can result in link rate oscillation, as evidenced by a high rate of link rate switches. We developed and evaluated a utilization-threshold policy and a time-out-threshold policy to eliminate oscillations and it was shown that these policies are successful in eliminating rate oscillations for smooth traffic. In summary, the utilization-threshold policy should be used if the complexity of counting packet arrivals is acceptable. If this complexity is not acceptable, the time-out-threshold policy should be used. In no case should the dual-threshold policy be used due to its inherent oscillation given smooth traffic.

ALR may enable significant energy savings within LAN switches and Ethernet-connected devices by allowing internal components to operate at a lower clock rate when the Ethernet link data rate is reduced. It is future work to investigate how these deeper energy savings could be achieved. Support for more than two data rates with an ALR policy is likely feasible if multiple utilization thresholds are defined. Investigation of multiple data rate ALR policies is a subject for future work. Also a subject for future work is to study if ALR has effects on higher layer protocols such as TCP.

ALR is currently a key component of a new Energy Efficient Ethernet (EEE) task force (IEEE 802.3az) in IEEE 802.3 [20]. The main focus of the task force is Rapid PHY Selection (RPS), which is the name now given to the mechanism part of ALR that switches the link rate. Policies such as those studied in this paper are outside the scope of IEEE 802.3 and remain an area for research and innovation and possible standardization outside of IEEE 802.3. ALR has significant potential for achieving considerable energy savings in the near future as part of the IEEE 802.3 Ethernet standard and products that conform to the standard.

Acknowledgments

The authors would like to thank Bob Grow of Intel Corp. for his helpful comments and discussions on ALR. The authors would also like to thank Mike Bennett for his leadership of the IEEE 802.3az (Energy Efficient Ethernet) task force. This material is based on work supported by the US National Science Foundation under Grant No. 0520081.

References

[1] L. Benini, A. Bogliolo, G. Paleologo, and G. De Micheli, "Policy Optimization for Dynamic Power Management," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 6, pp. 813-833, June 1999.

[2] "Broadcom BCM5701 10/100/1000BASE-T Controller Product Brief," http://www.broadcom.com/collateral/pb/5701-PB10-R.pdf, 2005.

[3] "Carbon Dioxide Emissions from the Generation of Electric Power in the United States," US Dept. of Energy and the Environmental Protection Agency, http://www.eia.doe.gov/cneaf/electricity/page/co2_report/co2report.html, July 2000.

[4] E. Chong and W. Zhao, "Performance Evaluation of Scheduling Algorithms for Imprecise Computer Systems," *J. Systems and Software*, vol. 15, no. 3, pp. 261-277, July 1991.

[5] K. Christensen, C. Gunaratne, B. Nordman, and A. George, "The Next Frontier for Communications Networks: Power Management," *Computer Comm.*, vol. 27, no. 18, pp. 1758-1770, Dec. 2004.

[6] E.-Y. Chung, L. Benini, A. Bogliolo, Y.-H. Lu, and G. De Micheli, "Dynamic Power Management for Nonstationary Service Requests," *IEEE Trans. Computers*, vol. 51, no. 11, pp. 1345-1361, Nov. 2002.

[7] M. Crovella, M. Harchol-Balter, and C. Murta, "Task Assignment in a Distributed System: Improving Performance by Unbalancing Load," *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 26, no. 1, pp. 268-269, June 1998.

[8] "EU Stand-by Initiative, Minutes of the Third Meeting on Energy Consumption of Broadband Communication Equipment and Networks," European Commission DG JRC, Ispra, http://energyefficiency.jrc.cec.eu.int/html/standby_initiative.htm, Nov. 2005.
[9] A. Field, U. Harder, and P. Harrison, "Network Traffic Behaviour in Switched Ethernet Systems," *Performance Evaluation,* vol. 58, no. 2, pp. 243-260, 2004.
[10] R. Gebhard, "A Queueing Process with Bilevel Hysteretic Service-Rate Control," *Naval Research Logistics Quarterly,* vol. 14, pp. 55-68, 1967.
[11] G. Ginis, "Low Power Modes for ADSL2 and ADSL2+," SPAA021, Broadband Comm. Group, Texas Instruments, Jan. 2005.
[12] S. Gochman et al., "The Intel Pentium M Processor: Microarchitecture and Performance," *Intel Technology J.,* vol. 7, no. 2, pp. 21-36, May 2003.
[13] R. Golding, P. Bosch, and J. Wilkes, "Idleness Is Not Sloth," Technical Report HPL-96-140, Hewlett-Packard Laboratories, Oct. 1996.
[14] D. Gross and C. Harris, *Fundamentals of Queueing Theory,* third ed. John Wiley & Sons, 1998.
[15] C. Gunaratne and K. Christensen, "Ethernet Adaptive Link Rate: System Design and Performance Evaluation," *Proc. 31st IEEE Conf. Local Computer Networks,* pp. 28-35, Nov. 2006.
[16] C. Gunaratne, K. Christensen, and B. Nordman, "Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed," *Int'l J. Network Management,* vol. 15, no. 5, pp. 297-310, Sept./Oct. 2005.
[17] C. Gunaratne, K. Christensen, and S. Suen, "Ethernet Adaptive Link Rate (ALR): Analysis of a Buffer Threshold," *Proc. IEEE Global Telecomm. Conf.,* Nov. 2006.
[18] M. Gupta and S. Singh, "Greening of the Internet," *Proc. ACM SIGCOMM '03,* pp. 19-26, Aug. 2003.
[19] M. Gupta, S. Grover, and S. Singh, "A Feasibility Study for Power Management in LAN Switches," *Proc. 12th IEEE Int'l Conf. Network Protocols,* pp. 361-371, Oct. 2004.
[20] IEEE 802.3 Energy Efficient Ethernet Study Group, http://grouper.ieee.org/groups/802/3/eee_study/index.html, 2006.
[21] IEEE 802.3 LAN/MAN CSMA/CD Access Method, http://standards.ieee.org/getieee802/802.3.html, 2006.
[22] ITU Recommendation G.992.3: Asymmetric Digital Subscriber Line Transceivers 2 (ADSL2), http://www.itu.int/rec/T-REC-G.992.3/en, 2006.
[23] ITU Recommendation G.992.5: Asymmetric Digital Subscriber Line (ADSL) Transceivers, http://www.itu.int/rec/T-REC-G.992.5/en, 2006.
[24] ITU Recommendation G.993.1: Very High Speed Digital Subscriber Line Transceivers, http://www.itu.int/rec/T-REC-G.993.1/en, 2006.
[25] V. Kulkarni, *Modeling, Analysis, Design, and Control of Stochastic Systems,* Springer, 1999.
[26] B. Nordman and K. Christensen, "Reducing the Energy Consumption of Network Devices," Tutorial presented at the July 2005 IEEE 802 LAN/MAN Standards Committee Plenary Session, http://www.csee.usf.edu/~christen/energy/pubs.html, July 2005.
[27] B. Nordman and A. Meier, "Energy Consumption of Home Information Technology," Technical Report LBNL-53500, Energy Analysis Dept., Lawrence Berkeley Nat'l Laboratory, July 2004.
[28] A. Odlyzko, "Data Networks Are Lightly Utilized, and Will Stay That Way," *Rev. Network Economics,* vol. 2, no. 3, pp. 210-237, Sept. 2003.
[29] Z. Ren, B. Krogh, and R. Marculescu, "Hierarchical Adaptive Dynamic Power Management," *IEEE Trans. Computers,* vol. 54, no. 4, pp. 409-420, Apr. 2005.
[30] J. Roberson, C. Webber, M. McWhinney, R. Brown, M. Pinckard, and J. Busch, "After-Hours Power Status of Office Equipment and Inventory of Miscellaneous Plug-Load Equipment," Technical Report LBNL-53729, Energy Analysis Dept., Lawrence Berkeley Nat'l Laboratory, Jan. 2004.
[31] K. Roth, F. Goldstein, and J. Kleinman, *Energy Consumption by Office and Telecommunications Equipment in Commercial Buildings, Volume I: Energy Consumption Baseline.* Arthur D. Little Reference No. 72895-00, Jan. 2002.
[32] K. Roth, R. Ponoum, and F. Goldstein, *U.S. Residential Information Technology Energy Consumption in 2005 and 2010.* TIAX Reference No. D0295, Mar. 2006.
[33] H. Schwetman, "CSIM19: A Powerful Tool for Building System Models," *Proc. 33rd Winter Simulation Conf.,* pp. 250-255, Dec. 2001.
[34] M.B. Srivastava, A.P. Chandrakasan, and R.W. Brodersen, "Predictive System Shutdown and Other Architectural Techniques for Energy Efficient Programmable Computation," *IEEE Trans. Very Large Scale Integration (VLSI) Systems,* vol. 4, no. 1, pp. 42-55, Mar. 1996.
[35] United States Energy Information Administration, *Electric Power Monthly,* Table 5.3, http://www.eia.doe.gov/cneaf/electricity/epm/epm_sum.html, June 2007.
[36] "US EPA Energy Star Program Requirements for Computers: Version 4.0," http://www.energystar.gov/index.cfm?c=revisions.computer_spec, 2006.
[37] C. Webber, J. Roberson, R. Brown, C. Payne, B. Nordman, and J. Koomey, "Field Surveys of Office Equipment Operation Patterns," Technical Report LBNL-46930, Energy Analysis Dept., Lawrence Berkeley Nat'l Laboratory, Sept. 2001.
[38] K. Yoshigoe and K. Christensen, "A Parallel-Polled Virtual Output Queued Switch with a Buffered Crossbar," *Proc. IEEE Workshop High Performance Switching and Routing,* pp. 271-275, May 2001.

Chamara Gunaratne received the BSc (engineering) degree in computer science and engineering from the University of Moratuwa, Sri Lanka, in 2000 and the PhD degree from the Department of Computer Science and Engineering at the University of South Florida in December 2006. His research interests include dynamic power management, performance evaluation, and capacity planning of computer networks. He has been a student member of the IEEE since 2003.

Ken Christensen received the PhD degree in electrical and computer engineering from North Carolina State University in 1991. He is a professor in the Department of Computer Science and Engineering at the University of South Florida. His research interest is in the performance evaluation of computer networks. He has served as the general chair and program chair (twice) of the IEEE Conference on Local Computer Networks. He is currently an associate editor for the *International Journal of Network Management*, published by John Wiley & Sons. He is a licensed professional engineer in the state of Florida, a senior member of the IEEE, and a member of the ACM and the ASEE.

Bruce Nordman received the BA degree in architecture and the MA degree in energy and resources from the University of California, Berkeley, in 1984 and 1990, respectively. He is a researcher with the Environmental Energy Technologies Division at the Lawrence Berkeley National Laboratory (LBNL). He has worked on energy efficiency in electronics at LBNL since the early 1990s, doing research for the EPA Energy Star program, California Energy Commission, and US Department of Energy.

Stephen Suen received the PhD degree in mathematics from the University of Bristol, England, in 1985. He is an associate professor in the Department of Mathematics and Statistics at the University of South Florida. His research interests are in the area of probabilistic combinatorics.