Optimized eeeBond: Energy Efficiency with non-Proportional Router Network Interfaces

Niklas Carlsson
Linköping University, Sweden
niklas.carlsson@liu.se

ABSTRACT

The recent Energy Efficient Ethernet (EEE) standard and the eBond protocol provide two orthogonal approaches that allow significant energy savings on routers. In this paper we present the modeling and performance evaluation of these two protocols and a hybrid protocol. We first present eeeBond, pronounced "triple-e bond", which combines the eBond capability to switch between multiple redundant interfaces with EEE's active/idle toggling capability implemented in each interface. Second, we present an analytic model of the protocol performance, and derive closed-form expressions for the optimized parameter settings of both eBond and eeeBond. Third, we present a performance evaluation that characterizes the relative performance gains possible with the optimized protocols, as well as a trace-based evaluation that validates the insights from the analytic model. Our results show that there are significant advantages to combine eBond and EEE. The eBond capability provides good savings when interfaces only offer small energy savings when in short-term sleep states, and the EEE capability is important as short-term sleep savings improve.

Keywords

Energy Efficiency, EEE, eBond, Adaptive Link Rate, Energy Proportional Computing, Router Performance

1. INTRODUCTION

High energy costs associated with the operation of network equipment have prompted the development of energy efficient policies and techniques for router management [1–3]. This desire has been further compounded by the high CO_2 emissions associated with non-green energy sources and an expectation of increasing energy prices.

Energy proportionality has been expressed as a desirable energy target [4], suggesting that the energy usage of a system should be proportional to the system utilization. As Internet routers typically are over provisioned, serve highly diurnal and time varying workloads, and often operate at low utilization, there should be substantial room for energy savings. However, due to hardware limitations, the energy consumption of active router interfaces are not energy proportional, and the full energy savings are therefore often difficult to achieve in practice. For this reason, protocols and policies have been proposed to make the best possible use of the existing hardware.

Two fundamental and promising approaches to scale the energy usage based on the current traffic load is to save energy by either (i) toggle between parallel, redundant, and heterogeneous interfaces [5], or (ii) toggle each interface between an active high-power mode and a low-power idle mode, during which some interface components are put to temporary sleep [3]. eBond [5] takes the first approach. It uses redundant heterogeneous links and Ethernet's bonding feature to toggle between which interface is used. When the router is lightly loaded a low-bandwidth link (with lower energy usage) is used, allowing the regular high-bandwidth link (with higher energy usage) to be turned off. In contrast, the recent Energy Efficient Ethernet (EEE) [3] standard takes the second approach. EEE allows an individual interface to save energy by switching between a low-power idle mode and an active high-power mode. Both eBond and EEE can allow significant energy savings, but both come with shortcomings.

In this paper we make three contributions towards improving and understanding the energy savings of routers. First, we present eeeBond (pronounced "triple-e bond"), which combines eBond and EEE into a simple hybrid protocol. As shown in Table 1, eeeBond combines the eBond capability to toggle between multiple heterogeneous redundant interfaces with EEE's active/idle toggling capability implemented in each interface. Second, we present a unified analytic model of the performance of these protocols and derive closed-form expressions and explicit conditions for the optimized parameter settings of both eBond and eeeBond. Using our model we analyze and discuss the energy saving tradeoffs in both existing and future systems.

Third, we present a performance evaluation that characterizes the performance gains possible with EEE and the optimized versions of both eBond and eeeBond. Our results show that there are significant advantages to combine eBond and EEE. For current technology that often see small energy savings in short-term sleep states, the eBond capability provides most of the energy savings, whereas the EEE component (especially if combined with eBond, as in eeeBond) provides great improvements when short-term sleep states would allow greater energy savings. The energy sav-

ICPE'16, March 12–18, 2016, Delft, Netherlands.

DOI: http://dx.doi.org/10.1145/2851553.2851564

Table 1: Protocol taxonomy

	Always active	Active/idle toggling
Single interface	Naive/default	EEE [3]
Multi interface	eBond [5]	eeeBond

ings with eBond, achieved through router management, are important as it is likely to be many years before we have fully proportional router hardware on the market, and the significant additional savings using eeeBond when short-term sleep states allow greater energy savings are encouraging for future systems. The conclusions based on our analytic models are complemented with a trace-based evaluation that validates the insights from the analytic model.

The remainder of the paper is organized as follows. Sections 2 and 3 presents an overview of the protocols considered in this paper, including eeeBond, and our system model, respectively. We then present our protocol optimizations (Section 4) and policy evaluation (Section 5), before concluding the paper with a discussion of related work (Section 6) and our conclusions (Section 7).

2. PROTOCOL OVERVIEW

The Energy Efficient Ethernet (EEE) [3] standard assumes a Lower Power Idle (LPI) mode and a (high-power) active mode, and defines the signaling that is required between the transmitter and the receiver when the former toggles back-and-forth between the two modes. Unfortunately, today's hardware does not allow EEE to be energy proportional. First, the interfaces often consume a significant amount of energy even when in sleep mode [5–7]. Second, there are non-negligible activation times and energy costs associated with activating an interface. For example, to achieve the suggested wakeup time (equal to the transmission time of the maximum length packet [6]) typically very few circuits in the physical layer can be turned off during the idle mode, resulting in only modest energy savings. As these hardware technologies continue to improve, and the energy usage in sleep states decrease, the expectation is that advanced hardware technologies will allow greater energy savings (up to 80%) [6].

An orthogonal approach that does not require such hardware improvements, is to leverage the use of redundant interfaces to allow one or more interfaces to go into deep sleep. As long as at least one sufficiently dimensioned interface is active, the router should be able to serve traffic demands.

This is the approach taken by eBond [5]. With eBond, the bonding protocol available and implemented in most routers is made energy-aware, such as to allow energy-aware switching between redundant heterogeneous links. For example, a low-bandwidth link can be used when the router is lightly loaded, allowing the regular high-bandwidth link to be turned off. By adapting which interface is active the capacity and energy usage can be tuned based on current traffic load.

Naturally, considering a single interface, the deep sleep modes used with eBond typically allow much greater energy savings, but comes at the cost of much longer activation times (than the sleep modes used by EEE). Therefore protocols switching between multiple redundant interfaces, must typically operate at a longer time granularity than the granularity at which EEE operates.

Motivated by eBond and EEE operating at different time scales, this papers considers a simple hybrid generalization that we call eeeBond. With eeeBond, routers would have the flexibility to both (i) switch between interfaces with different capacity and energy usage, and (ii) toggle the currently used interface between active and idle mode. We expect that an eeeBond system would use energy-aware bonding (putting some interfaces to long-term deep sleep) at larger time granularity, and active/idle toggling (to/from the short-term sleep state) at a finer time granularity. For example, interface selection can be based in diurnal long-term variations in traffic volumes and active/idle toggling can be used to take into account energy saving opportunities due to short-term variations in traffic intensity.

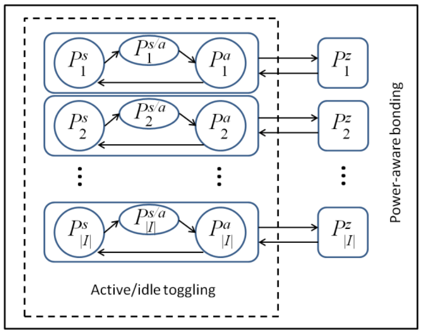


Figure 1: Router model and power states for each interface.

3. SYSTEM MODEL

We consider a single router and compare policies that differ based on their capability to (i) toggle between multiple redundant links (bonding as in eBond [5]) and (ii) perform active/idle toggling (as in EEE [3]). Table 1 summarizes the four resulting candidate protocols.

- **Naive baseline:** Single interface policy that does not attempt to leverage any low-power modes.
- **EEE using single interface:** Single interface policy that uses active/idle toggling to reduce energy usage.
- **eBond:** Multi-interface policy that use energy-aware bonding to save energy.
- **eeeBond:** Multi-interface policy that use both energy-aware bonding and active/idle toggling.

Naturally, all protocols are special cases of eeeBond, and only differ in which of the two types of energy saving capabilities are implemented. To compare the relative importance and tradeoffs associated with these energy saving capabilities, we present a unified model that captures both these aspects of eeeBond and the other protocols.

3.1 General Model

Figure 1 illustrates our basic router model. For the purpose of our discussion and analysis, consider a router with $|\mathcal{I}|$ redundant interfaces, where $\mathcal{I}$ is the set of such interfaces.

Energy usage: We will consider a basic hardware model in which each interface can be in one of four power states:

- An *active* high-power state with an average power usage P_i^a in which the interface operates at full link capacity μ_i.

- A low-power *short-term (light) sleep* state with an average power usage P_i^s, which allows the interface to quickly enter the active state.
- A short *setup period* Δ_i during which the interface is activated from the short-term (light) sleep period. The average power usage during this time period is $P_i^{s/a}$.
- A low-power *long-term deep sleep* state with an average power usage P_i^z, which allows bigger energy savings than short-term sleep (i.e., $P_i^z < P_i^s$) but that require longer activation periods, and hence only can be used at coarser time granularity.

Of course, in a real system, there would also be a state bringing the interface from the active to the short-term sleep state, as well as transition states bringing the interface in and out of the deep sleep state. However, for the purpose of the analysis we do not include these states. The energy consumed in the first case can easily be accounted by adjusting the $P_i^{s/a}$ term (as there is always a corresponding activation period). For the latter case, we note that the bonding policies considered here are expected to operate at a much longer time granularity and the system therefore would only be in these transition states for a very small fraction of the total system time.

Furthermore, motivated by the above time granularity differences, we assume that, at any given point in time, one interface is responsible for the current traffic over the link. This interface is in one of the first three states (i.e., in the active, short-term sleep, or in the interface setup period) shown on left-hand side of Figure 1. All other interfaces are assumed to be in the long-term deep sleep state (right-hand side of Figure 1).

Consider the system's power usage measured over a long time period, and let the probabilities q_i^a, q_i^s, $q_i^{s/a}$, and q_i^z represent the probability of observing interface i in each of these four states (equal to the fraction of time the system would spend in each state if measured for a very long time). With this notation, the average power usage can be calculated as:

$$P_{\mathcal{I}} = \sum_{i \in \mathcal{I}} \left[q_i^a P_i^a + q_i^s P_i^s + q_i^{s/a} P_i^{s/a} + q_i^z P_i^z \right]. \qquad (1)$$

Hardware comparison: For simple head-to-head policy comparison under both current and future hardware systems, we use system parameters (c_i, g_i) to capture the relative energy usage between the different states of an interface and the parameter x to capture the relative energy scaling seen between the heterogeneous interfaces' energy usage.

First, we use a constant c_i ($0 \leq c_i \leq 1$) to capture the energy savings ratio $c_i = \frac{P_i^s}{P_i^a}$, of the power usage in the short-term sleep and active mode, respectively. While current systems often have a ratio between 0.8 and 1, future systems may be able to achieve much smaller ratios (e.g., 0.2) [6]. In the ideal case $c_i = 0$. Second, we define $g_i = \frac{P_i^{s/a}}{P_i^a}$ as the ratio between the power usage during the active state and during the setup period. (For most cases, we assume $g_i \approx 1$.) Finally, to allow a wide range of scaling behaviors (and to accommodate for potential future energy trends, for example) we assume that $P_i^a = f(\mu_i)$, where $f(\mu_i) = P_0^a(\frac{\mu_i}{\mu_0})^x$ and P_0^a corresponds to the power usage for a reference system with service rate μ_0. We note that this function is linear when $x = 1$, sublinear when $x < 1$, and superlinear when $x > 1$. This model and model parameter is used to capture how the expected active power P_i^a differs between interfaces. While the energy usage clearly will vary significantly from implementation to implementation and likely will differ significantly in magnitude from the most efficient technology one decade to the most effective technology during the next decade, we expect that the current power usage typically will scale superlinearly ($x > 1$) with the service rate μ_i of the interfaces.

General per-interface delay: We extend the basic router model developed and validated using real traffic by Hohn et al. [8]. Motivated by current state of the art, only transmission delays and queueing delays on the outgoing interface are considered. Assuming a First-In-First-Out (FIFO) queueing policy and infinite buffer size (motivated by the low-loss scenario and line cards often able to accommodate up to 500 ms worth of traffic) the delay w_k experienced by the k^{th} packet of *active* router interface i with service rate μ_i is then

$$w_k = [w_{k-1} - (t_k - t_{k-1})]^+ + \frac{l_k}{\mu_i}, \qquad (2)$$

where $[y]^+ = \max(y, 0)$, and t_k is the arrival time of the k^{th} packet of length l_k. For additional details, motivation, and validation of the model and these assumptions, we refer the interested reader to the original paper [8].

To extend this model to the case in which an interface is allowed to be in short-term sleep mode whenever there are no packets to serve, we must take into account the time Δ_i it takes to activate the link from the short-term sleep mode when a new packet arrives. The delays of such a policy can be modeled as follows:

$$w_k = \begin{cases} \Delta_i + \frac{l_k}{\mu_i}, & \text{if } t_k > t_{k-1} + w_{k-1} \\ w_{k-1} + t_{k-1} - t_k + \frac{l_k}{\mu_i}, & \text{otherwise.} \end{cases} \qquad (3)$$

3.2 Steady-state Model

We next derive closed-form expressions for the expected waiting times and probabilities to be in each of the operation states. For this analysis, we consider a system in steady state, as often observed over shorter periods, for example [9]. Assuming that packets arrive according to a Poisson process (i.e., independent and exponentially distributed packet inter-arrival times) and the size of packets are mutually independent, we model the system as a $M/G/1(E, SU)$ queue with exhaustive service, multiple vacation periods, and setup time [10–13]. Under this model, the interface remains active serving packets (with expected service time $E[S_i]$) as long as there is at least one packet waiting to be served, and then goes on a "vacation" when there is no packet(s) to serve. When a new packet arrives to the interface, a setup time Δ_i is required the packet can be served. We also leverage the PASTA property that Poisson arrivals see time averages [14]. Given a packet arrival rate λ, the expected waiting time $\overline{W_i}$ for interface i can then be calculated as:

$$\overline{W_i} = E[w_k|i] = \frac{\lambda E[S_i^2]}{2(1-\rho_i)} + \frac{2E[\Delta_i] + \lambda E[\Delta_i^2]}{2(1+\lambda E[\Delta_i])}, \qquad (4)$$

where $E[S_i^2]$ is the expected service time squared, ρ_i is the expected interface utilization when interface i is active, and $E[\Delta_i]$ and $E[\Delta_i^2]$ are the expected setup times and setup-times squared, respectively. Note that the second term approaches zero as $\Delta_i \to 0$, and the expected waiting time therefore becomes equal to that of a regular $M/G/1$ queue (i.e., the first term in equation (4)) in this case.

To allow comparison of interfaces operating at different service rate, we break out the interface dependent service rates from the above expression for the expected waiting time $\overline{W_i}$. To do so, we use the following equalities: $E[S_i^2] = \frac{E[l_k^2]}{\mu_i^2}$, $\rho_i = \lambda \frac{E[l_k]}{\mu_i}$, $E[\Delta_i] = \frac{\max_k l_k}{\mu_i}$, and $E[\Delta_i^2] = \frac{\max_k l_k^2}{\mu_i^2}$. The first two equalities are true in general, whereas the last two are motivated by the EEE specifications which suggest that the setup time should be equal to the processing time of the largest packets; i.e., $\Delta_i = \max_k l_k/\mu_i$. With these observations, we can rewrite the expected waiting time as:

$$\overline{W_i} = \frac{\lambda \frac{E[l_k^2]}{\mu_i^2}}{2(1 - \lambda \frac{E[l_k]}{\mu_i})} + \frac{2 \frac{\max_k l_k}{\mu_i} + \lambda \frac{\max_k l_k^2}{\mu_i^2}}{2(1 + \lambda \frac{\max_k l_k}{\mu_i})}. \tag{5}$$

To evaluate the energy usage under steady state conditions, we next calculate the state probabilities q_i^a, q_i^s, $q_i^{s/a}$, and q_i^z. Note that q_i^z simply is the long-term probability not being in an active/idle-mode state (using EEE, for example). Consider therefore the probabilities when in such active/idle mode. In this case, the busy probability can be calculated as

$$\frac{q_i^a}{1 - q_i^z} = \rho_i = \lambda \frac{E[l_k]}{\mu_i}, \tag{6}$$

the idle probability can be calculated as

$$\frac{q_i^s}{1 - q_i^z} = \frac{1 - \rho_i}{1 + \lambda E[\Delta_i]} = \frac{1 - \lambda \frac{E[l_k]}{\mu_i}}{1 + \lambda \frac{\max_k l_k}{\mu_i}}, \tag{7}$$

and finally, the setup probability can be calculated as

$$\frac{q_i^{s/a}}{1 - q_i^z} = \frac{\lambda(1 - \rho) E[\Delta_i]}{1 + \lambda E[\Delta_i]} = \frac{\lambda(1 - \lambda \frac{E[l_k]}{\mu_i}) \frac{\max_k l_k}{\mu_i}}{1 + \lambda \frac{\max_k l_k}{\mu_i}}. \tag{8}$$

Here, $\rho_i = \lambda E[S] = \lambda \frac{E[l_k]}{\mu_i}$ is the utilization and $1+\lambda E[\Delta_i] = 1+\lambda \frac{\max_k l_k}{\mu_i}$ is the expected number of arrivals during an idle period. These probabilities can now be used with equation (1) to calculate the average power usage. For example, separating the power usage for the time period interfaces i is used and inserting the above conditional probabilities we obtain:

$$\frac{P_i - q_i^z P_i^z}{1 - q_i^z} = \left[\rho_i P_i^a + \frac{1 - \rho_i}{1 + \lambda E[\Delta_i]} P_i^s + \frac{\lambda(1 - \rho) E[\Delta_i]}{1 + \lambda E[\Delta_i]} P_i^{s/a}\right]$$
$$= \left[\lambda \frac{E[l_k]}{\mu_i} P_i^a + \frac{1 - \lambda \frac{E[l_k]}{\mu_i}}{1 + \lambda \frac{\max_k l_k}{\mu_i}} P_i^s + \frac{\lambda(1 - \lambda \frac{E[l_k]}{\mu_i}) \frac{\max_k l_k}{\mu_i}}{1 + \lambda \frac{\max_k l_k}{\mu_i}} P_i^{s/a}\right]. \tag{9}$$

Equation (9) captures the energy usage of an interface using EEE as used by the interface not in low-power deep-sleep mode with eeeBond. We will use equations (5) and (9) when selecting which interface to keep active.

4. PROTOCOL OPTIMIZATION

Both the energy-delay tradeoff and the optimal policies of eBond and eeeBond differ. Definition 1 defines what we mean with an optimal policy, and in the following subsections we define the optimal eBond and eeeBond policies, and use our analytic model to provide insights to their characteristics.

Definition 1. *The optimal policy always picks the interface with the lowest power usage, conditioned on also having an average waiting time W less than or equal to some threshold W^*. When no such interface exists, the policy picks the interface with the shortest expected waiting time W.*

4.1 Optimized eBond

Theorem 1. *Given an average target waiting time W^* and an estimated packet inter arrival rate λ, the optimal eBond policy always picks the interface with the lowest service rates μ_i that can support a packet arrival rate*

$$\lambda \leq \lambda_i^* = \frac{2(W^* - E[S_i])}{E[S_i^2] + 2E[S_i](W^* - E[S_i])}, \tag{10}$$

where $E[S_i] = \frac{E[l_k]}{\mu_i}$ and $E[S_i^2] = \frac{E[l_k^2]}{\mu_i^2}$.

Proof. (Theorem 1) First, the waiting time (for this M/G/1 queueing system without vacation periods) is monotonically non-decreasing. Second, we show that the energy usage of an interface with lower service rate μ_i always consumes less energy. To see this, note that we in this case have $\frac{q_i^a}{1-q_i^z} = 1$, $\frac{q_i^s}{1-q_i^z} = 0$, and $\frac{q_i^{s/a}}{1-q_i^z} = 0$. With these observations, the power usage $\frac{P_i}{1-p_i^z} = P_0^a(\frac{\mu_i}{\mu_0})^x$, clearly is monotonically non-decreasing for $x \geq 0$. To see this note that: $\frac{dP_i}{d\mu_i} = \frac{d}{d\mu_i}[P_0^a(\frac{\mu_i}{\mu_0})^x] = xP_0^a \frac{\mu_i^{x-1}}{\mu_0^x} \geq 0$. Third, we show that the expected waiting times are non-decreasing. Taking the derivative of the average waiting time in a M/G/1 queue (without vacations)

$$W_i = \frac{\lambda E[S_i^2]}{2(1 - \rho_i)} + E[S_i] = \frac{\lambda \frac{E[l_k^2]}{\mu_i^2}}{2(1 - \lambda \frac{E[l_k]}{\mu_i})} + \frac{E[l_k]}{\mu_i} \tag{11}$$

with regards to the service rate, we get:

$$\frac{dW_i}{d\mu_i} = -\frac{E[S_i^2] \lambda E[l_k]}{2(1 - \rho_i)^2 \mu_i^2} - \frac{E[l_k]}{(1 - \rho_i)\mu_i^3} - \frac{E[l_k]}{\mu_i^2}, \tag{12}$$

which clearly is no greater than zero (as all three terms are negative) for all $0 \leq \rho_i \leq 1$. Fourth, with monotonic ordering of the energy usage and waiting times in terms of both μ_i and λ, we can obtain the threshold value λ_i^* by setting equation (11) equal to W^* and solving for λ_i^*. After minor reordering we obtain equation (10). This completes the proof. □

4.2 Optimized eeeBond

Before defining and proving the optimal eeeBond policy we first identify and prove five properties of eeeBond. These are defined in the following five lemmas.

First, note that the use of a setup period causes the average waiting time W_i to be a non-monotonic function that first decreases and then increase with the packet arrival rate λ. To see this, note that the waiting time for very low arrival rates approaches Δ_i as $\lambda \to 0$, is lower for intermediate arrival rates (for which many packets may arrive with only a single packet ahead of them in the queue[1]), and then increase

[1] Each such packet sees a conditioned waiting time equal to the residual service time, which is smaller than the expected service time $E[W_k|1] \leq E[S_i]$, and hence also smaller than the setup time $E[S_i] = \frac{E[l_k]}{\mu_i} \leq \frac{\max_k l_k}{\mu_i} = \Delta_i$.

again as the link utilization approaches one. Our first lemma formalizes these observations and defines conditions for (i) when the waiting times are monotonically non-decreasing and (ii) when the average waiting times W_i with an arrival rate λ always is lower than that of a baseline arrival rate λ^*.

LEMMA 1. *The expected waiting time W_i is a monotonically non-decreasing function of the arrival rate λ for the region in which $W_i \geq \Delta_i$, and for any $\lambda \leq \lambda^*$ for which $W_i^* = W_i(\lambda^*) \geq \Delta_i$, the waiting time $W_i(\lambda) \leq W_i(\lambda^*)$.*

PROOF. (Lemma 1) Consider the derivative of the expected waiting time:

$$\frac{dW}{d\lambda} = \frac{E[S_i^2]}{2(1-\rho_i)^2} - \frac{\Delta_i^2}{2(1+\lambda\Delta_i)^2}. \tag{13}$$

This function is negative for $\lambda < \frac{\Delta_i^2 - E[S_i^2]}{\Delta_i E[S_i^2] + \Delta_i^2 E[S_i]}$ and positive for $\frac{\Delta_i^2 - E[S_i^2]}{\Delta_i E[S_i^2] + \Delta_i^2 E[S_i]} < \lambda$. Let $\lambda^{**} = \frac{\Delta_i^2 - E[S_i^2]}{\Delta_i E[S_i^2] + \Delta_i^2 E[S_i]}$ define the arrival rate with the minimum waiting time. With $W(\lambda) \rightarrow \Delta_i$ as $\lambda \rightarrow 0$ and a single minimum, we know that the minimum waiting time $W_i^{**} = W_i(\lambda^{**}) \leq \Delta_i$ and there exists a $\lambda^{***} \geq \lambda^{**}$ for which $W_i(\lambda^{***}) = \Delta_i$. Clearly, for $0 \leq \lambda \leq \lambda^{***}$, we have $W_i(\lambda) \leq \Delta_i \leq W_i$, and for any larger packet arrival rates $\lambda^{***} \leq \lambda$ the function is monotonically non-decreasing. This completes the proof. □

LEMMA 2. *The expected waiting time W_i is a monotonically non-increasing function of the service rate μ_i.*

PROOF. (Lemma 2) This proof is relatively straight forward. First, note that the waiting time W_i of a $M/G/1(E, SU)$ queue can be broken up into a term $W_i^{M/G/1}$ that is independent of the setup time Δ_i and a second term $W_i^{\Delta_i}$ that depends on the setup time. As for any M/G/1 system, the first term is non-increasing. For the second term we substitute $\Delta_i = \frac{\max_k l_k}{\mu_i}$ and take the derivative with regards to μ_i:

$$\frac{dW_i^{\Delta_i}}{d\mu_i} = \frac{d}{d\mu_i}\left[\frac{2\Delta_i + \lambda\Delta_i^2}{2(1+\lambda\Delta_i)}\right] = \frac{-4\frac{\Delta_i}{\mu_i} - 4\lambda\frac{\Delta_i^2}{\mu_i} - 2\lambda^2\frac{\Delta_i^3}{\mu_i}}{4(1+\lambda\Delta_i)^2} \leq 0.$$

This function is non-positive, and hence both $W_i^{\Delta_i}$ and W_i must be non-increasing with μ_i. □

LEMMA 3. *Given a target delay $W^* \geq \Delta_i$, unless there does not exist any interface with higher service rate, the optimal policy never picks a low-power interface with service rate μ_i when the packet arrival rate λ exceeds an upper bound*

$$\lambda_i^u = \frac{-a_1 + \sqrt{a_1^2 - 4a_2a_0}}{2a_2}, \tag{14}$$

where $a_2 = \Delta_i E[S_i](2W^* - \Delta_i) + \Delta_i E[S_i^2]$, $a_1 = E[S_i^2] + 2E[S_i](W^* - \Delta_i) + \Delta_i(\Delta_i - 2W^*)$, and $a_0 = 2(\Delta - W^*)$.

PROOF. (Lemma 3) This proof follows directly from Lemmas 1 and 2. As per the monotonicity property in Lemma 1, there must exist an arrival rate λ^u such that the waiting time W_i of interface i is greater than W^* for all $\lambda > \lambda_i^u$. From Lemma 2 it also follows that in the case $\lambda > \lambda_i^u$ and $W_i \geq W^* \geq \Delta$, the waiting time of an interface with the same λ but higher service rate is no worse. Therefore, interface i should never be selected in this case. Setting equation (5) equal to W^* and rewriting the equation we obtain a second-order equation $a_2\lambda^2 + a_1\lambda + a_0 = 0$, with the parameters a_2, a_1 and a_0 defined as in the above lemma. While such an equation has two solutions, it is easy to show that only the solution above is positive and of consideration. To see this, note that the constraint $W^* \geq \Delta_i$ directly implies that $a_2 \geq 0$ and $a_0 \leq 0$. Therefore, $\sqrt{a_1^2 - 4a_2a_0} \geq -a_1$ and only the solution shown in the lemma is positive; completing our proof. □

LEMMA 4. *The expected power usage P_i is a monotonically non-decreasing function of the service rate μ whenever the relative energy scaling parameter x satisfies the condition:*

$$x \geq x^* = \frac{\Delta_i + E[S_i]}{G + \lambda H}, \tag{15}$$

where $G = c + (1+c)\Delta_i + (1-c)E[S_i]$, $H = \Delta_i(\Delta_i + (1-c)E[S_i])$, and $c = \frac{P_i^s}{P_i^a}$. Otherwise, the expected power usage P_i is a monotonically non-increasing function of μ.

PROOF. (Lemma 4) With the power usage during a "non-deep-sleep" period either being equal to P_i^a (active and transition mode) or P_i^s (sleep mode), we can rewrite the power usage as:

$$\begin{aligned}\frac{P_i}{1-q_i^z} &= \frac{1-\lambda E[S_i]}{1+\lambda\Delta_i}P_i^s + (1 - \frac{1-\lambda E[S_i]}{1+\lambda\Delta_i})P_i^a \\ &= \frac{c + \lambda\Delta_i + \lambda E[S_i](1-c)}{1+\lambda\Delta_i}P_i^a. \end{aligned} \tag{16}$$

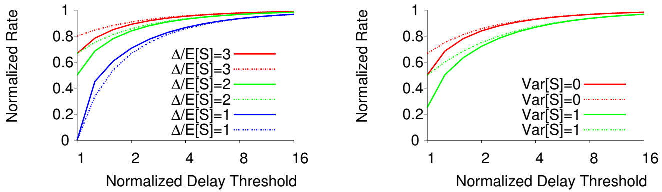
By identifying μ_i terms, we can rewrite this expression as $\frac{c+\frac{a}{\mu_i}}{1+\frac{b}{\mu_i}}P_i^a(\mu_i)$, where $a = \lambda^2\Delta_i + \lambda^2(1-c)E[S_i]$ and $b = \lambda\mu_i\Delta_i$. With $\frac{d}{d\mu_i}(\frac{c+\frac{a}{\mu_i}}{1+\frac{b}{\mu_i}}) = \frac{cb-a}{\mu_i^2(1+\frac{b}{\mu_i})^2}$ and $\frac{d}{d\mu_i}P_i^a(\mu_i) = \frac{x}{\mu_i}P_i^a$, we can now calculate the derivative

$$\begin{aligned}\frac{d}{d\mu_i}(\frac{P_i}{1-q_i^z}) &= \frac{cb-a}{\mu_i^2(1+\frac{b}{\mu_i})^2}P_i^a + \frac{c+\frac{a}{\mu_i}}{1+\frac{b}{\mu_i}}\frac{x}{\mu_i}P_i^a \\ &= \frac{cb(1+x) + a(x-1) + cx\mu_i + \frac{abx}{\mu_i}}{\mu_i^2(1+\frac{b}{\mu_i})^2}. \end{aligned} \tag{17}$$

Clearly, the derivative of this function is non-negative. As this function is non-positive when $x = 0$ (as $\frac{cb-a}{\mu_i^2(1+\frac{b}{\mu_i})^2} = \frac{-\lambda(\Delta_i+E[S_i])}{\mu_i(1+\frac{b}{\mu_i})^2} \leq 0$) and it is trivial to find positive values for larger x (e.g., for $x = 1$ the function is $\frac{c\mu_i+\frac{ab}{\mu_i}}{\mu_i^2(1+\frac{b}{\mu_i})^2} \geq 0$), there must therefore exist an x^* such that the function $\frac{P_i}{1-q_i^z}$ is monotonically non-decreasing function whenever $x \geq x^*$ and monotonically non-increasing otherwise. Setting equation (17) equal to zero, solving for x^*, and identifying terms, gives $x^* = \frac{A-cB}{cB+A+c+\frac{AB}{\mu_i}}$, where $A = \frac{a}{\lambda} = \lambda\Delta_i + \lambda(1-c)E[S_i]$ and $B = \frac{b}{\mu_i} = \lambda\Delta_i$. Finally, inserting the expressions for A and B and simplifying the expression (including identifying G and H), while isolating λ, completes the proof. □

Leveraging the Lemmas 1-4 we are now in a position to define and prove the optimal interface selection for eeeBond.

LEMMA 5. *Unless there does not exist another interface with higher service rate, the optimal policy never picks a low-rate, low-power interface with service rate μ_i (over an*



(a) Impact of $\frac{\Delta}{E[S_i]}$ (b) Impact of $\mathrm{Var}[S_i]$

Figure 2: Normalized threshold rates calculated per equations (10) and (14) for different example scenarios.

interface with higher service rate) when the packet arrival rate λ is less than a lower bound

$$\lambda_i^l = \frac{1}{H}(\Delta_i + E[S_i] - xG), \tag{18}$$

where G and H are defined as in Lemma 4.

Proof. (Lemma 5) The proof builds upon Lemmas 4 and 2. Lemma 4 implies that for a given arrival rate λ, the power usage will only be lower at the low-rate interface when $x \geq x^*$. As the targeted average waiting time conditioned on being no smaller than Δ, is a monotonically non-decreasing function (Lemma 2) of the service rate μ_i, there is therefore never an advantage in selecting the low-rate interface unless $x < x^*$. Now, taking the derivative of x^* (equation (15)), we note that the derivative $\frac{dx^*}{d\lambda} \leq 0$ is a non-positive function of λ. This shows that for any x (observed for the current technology), there exists a λ_i^l such that $x \geq x^*$ for all $\lambda \geq \lambda_i^l$. To find this λ_i^l we insert $x^* = x$ and $\lambda = \lambda_i^l$ in equation (15) and solve for λ_i^l. This completes the proof. □

Theorem 2. *Given a target waiting time $W^* \geq \max_i \Delta_i$ and arrival rate λ, the optimal eeeBond policy picks the lowest powered interface that satisfy both (i) $\lambda_i^l \leq \lambda$, and (ii) $\lambda \leq \lambda_i^u$, where λ_i^l and λ_i^u are given by equations (18) and (14), respectively. In the case no interface satisfies both constraints, the optimal policy picks the highest capacity interface.*

Proof. (Theorem 2) This theorem follows directly from Lemmas 3 and 5. Per these lemmas, equation (18) lower bounds the arrival rates for when an interface i is a candidate and equation (14) upper bounds the arrival rates for when interface i is a candidate. □

When applying Theorem 2 it is important to note that x typically is greater than 1 and λ_i^l therefore typically is non-positive. To see this, let us take a closer look at the lower bound (18) in Lemma 5. This expression is positive only when

$$x \leq \frac{\Delta_i + E[S_i]}{\Delta_i + E[S_i] + c(1 + \Delta_i - E[S_i])}. \tag{19}$$

With $\Delta_i \geq E[S_i]$ and $c \geq 0$, equation (19) is lower bounded by 1. Motivated by this observation, we focus on the upper bounds for eBond (equation (10)) and eeeBond (equation (14)). Figure 2 shows these two bounds as a function of the normalized delay threshold $\frac{W^*}{\Delta}$. Without loss of generality we use $E[S_i] = 1$ and measure the packet arrival rate λ in normalized units. With these normalized units, one time unit is equal to the average processing time of a packet, and the λ values shown in the figures are equal to the utilization of the interface at the point when it is better to switch to a higher capacity interface.

Table 2: Normalized power usage with diurnal model.

Scenario			Current $c = 0.8$			Future $c = 0.2$		
	$\overline{u}$	x	EEE	eBond	e^{3}B	EEE	eBond	e^{3}B
Two	0.5	1.2	0.95	0.76	**0.73**	0.76	0.76	**0.65**
	0.25	1.2	0.90	**0.49**	0.49	0.58	0.49	**0.38**
	0.125	1.2	0.87	0.44	**0.39**	0.44	0.44	**0.25**
	0.25	0.8	0.90	0.62	**0.60**	0.58	0.62	**0.47**
	0.25	2	0.90	**0.32**	0.34	0.58	0.32	**0.27**
Three	0.5	1.2	0.95	**0.74**	0.81	0.76	0.74	**0.68**
	0.25	1.2	0.90	**0.57**	0.59	0.58	0.57	**0.45**
	0.125	1.2	0.87	**0.43**	0.47	0.44	0.43	**0.30**
	0.25	0.8	0.90	**0.66**	0.67	0.58	0.66	**0.51**
	0.25	2	0.90	**0.45**	0.48	0.58	0.45	**0.36**

We note that the rate thresholds are greatest in the cases with (i) the largest difference between the processing time of the largest packets (Δ) and average packets ($E[S_i]$), and (ii) the smallest variance in the processing time ($\mathrm{Var}[S_i] = E[S_i^2] - E^2[S_i]$). This is to be expected as relatively small packets with small variations allow the system to operate at a higher utilization given a fixed delay threshold.

5. POLICY EVALUATION

5.1 Head-to-Head Comparison

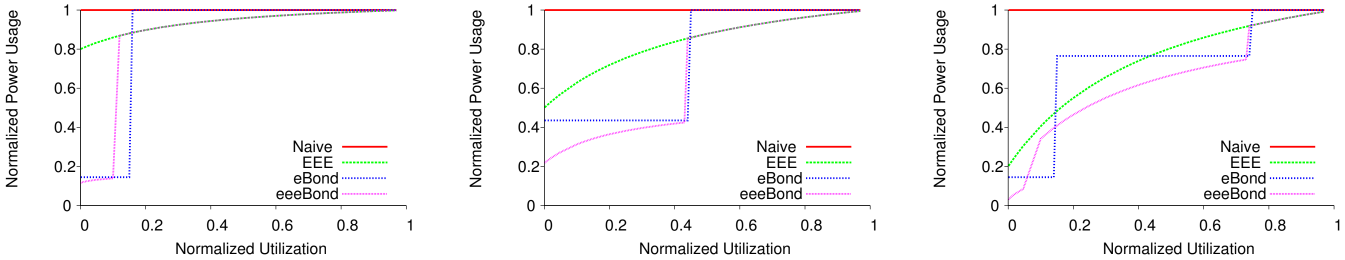
To understand the relative performance of the four policies outlined in Section 2, we have evaluated the power usage for a wide range of scenarios. Figure 3 shows three such example scenarios. Here, we have used the packet size distributions from both edge and core traces (cf. Table 3), different power ratios $c = \frac{P_i^s}{P_i^a}$, different number of interfaces, and different service rate ratios $\frac{\mu_2}{\mu_1}$ and $\frac{\mu_3}{\mu_1}$.

We note that energy savings in sleep state typically are small today (e.g., $c = 0.8$), but are expected to improve (e.g., $c = 0.2$) in the future. While both eBond and eeeBond can achieve substantial energy savings in all scenarios, these results show that eeeBond (and EEE) perhaps have the greatest benefits as energy savings in sleep state improve (smaller c).

An interesting observation is that there are regions where eBond performs better than basic eeeBond. In these regions it is better not to shut off the low-power interface. A further improved policy would therefore try to recognize these regions, allowing us to match the bottom line (either eBond or eeeBond) in each figure.

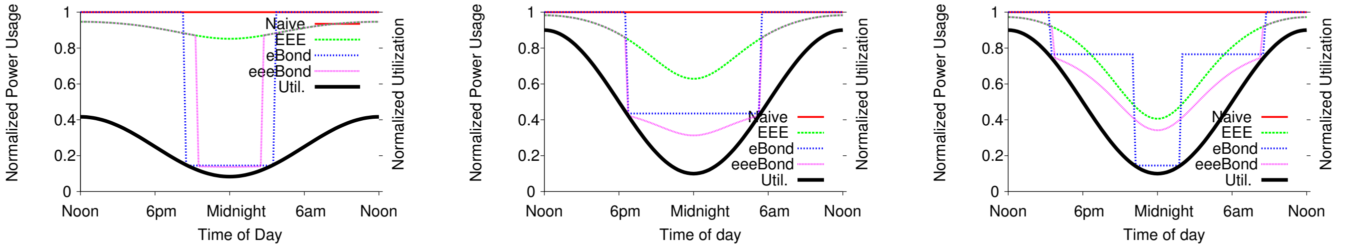
Figure 4 shows the power usage as a function of the time of day for three example workloads. In each case, the packet arrival rate is calculated using a sinusoidal. Motivated by the light load typically seen in edge networks and somewhat heaver load in core networks, we combine the packet sizes from an edge network with a utility function with averege utility $\overline{u} = 0.25$ and the packet sizes from a core network with a utility function with average utility $\overline{u} = 0.5$. For the edge example we use a max-min ratio of 5 and for the core cases we use a max-min ratio of 9. We can again see that there are regions where both eBond and eeeBond have their respective advantages, but that they typically both significantly outperform Naive and EEE. Only when c is very small does EEE compete with eBond, and in no case does it perform better than eeeBond.

Table 2 summarizes the average normalized power usage



(a) Edge, $\frac{\mu_2}{\mu_1} = 0.2, c = 0.8$ (b) Core, $\frac{\mu_2}{\mu_1} = 0.5, c = 0.5$ (c) Core, $\frac{\mu_2}{\mu_1} = 0.8$, $\frac{\mu_3}{\mu_1} = 0.2, c = 0.2$

Figure 3: Energy usage under example scenarios. Analytic results based on packet size statistics from edge and core traces.



(a) Edge, $\frac{\mu_2}{\mu_1} = 0.2, c = 0.8$ (b) Core, $\frac{\mu_2}{\mu_1} = 0.5, c = 0.5$ (c) Core, $\frac{\mu_2}{\mu_1} = 0.8$, $\frac{\mu_3}{\mu_1} = 0.2, c = 0.2$

Figure 4: Time-of-day comparison using diurnal request rates. Switching instances calculated for optimized protocols (Section 4).

Table 3: Packet size statistics.

	$E[l_k]$	$E[l_k^2]$	$\max_k l_k$
Edge, incoming	641.817	861,414	1,514
Edge, outgoing	589.725	764,668	1,514
Core, dirA	910.32	1,271,850	1,514
Core, dirB	545.295	729,769	1,514

(across a full day) for different average utilization ($\overline{u}$) and scaling factor x for the two interface scenario in Figure 4(b) and the three interface scenario in Figure 4(c), respectively. Again, eBond is competitive (and often best) when there are little energy savings from putting the interface to sleep (large c), and eeeBond (e^{3}B) is by far the best when the sleep savings are greater (small c). Clearly, eeeBond and similar hybrid protocols that combine both eBond functionality and the EEE protocol, may become increasingly beneficial as sleep savings become greater (smaller c).

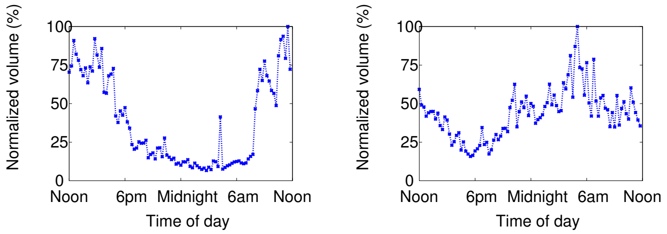
5.2 Trace-based evaluation

We have also evaluated the protocols using trace-based simulations. We use core traces collected at a core router (labeled *samplepoint-F*) connected to a trans-pacific link [15] and edge traces collected at an edge router (labeled Waikato VIII) of a university network [16]. Both traces were collected over a 24-hour period on January 2, 2013.

Table 3 summarizes the packet size information for the traces and Figure 5 shows the normalized traffic volume for each 15-minute period (with the peak volume that day normalized at 100%) for two of the traces. A closer look at these traces reveal that packet sizes for these traces are binomial in nature, with most packets being either small (less than 100 bytes) or large (1400-1500 bytes), and the relative fractions highly dependent on the direction.

Table 4 shows example results for the four example traces when using the same two-interface scenario as for the analytic results. For this analysis, the traces are broken up into 15 minute intervals and the policies are applied on a per-15 minute granularity. Interesting future work could consider adaptive policies that apply threshold-based rules within a moving window, for example. The lower power usage for the traces is in part due to the links being lightly utilized. The results do, however, confirm that our conclusions regarding the protocols relative performance with different sleep-saving efficiency (c) are consistent also for real traces.

Finally, we note that our model easily can be extended to more closely match the traffic seen in the traces. Figure 6 shows example results from our basic model and an extended model (omitted due to lack of space), based on an $M^X/G/1(E, SU)$ system [13]. Here, the packet size statistics from a 15-minute edge trace of the incoming traffic be-

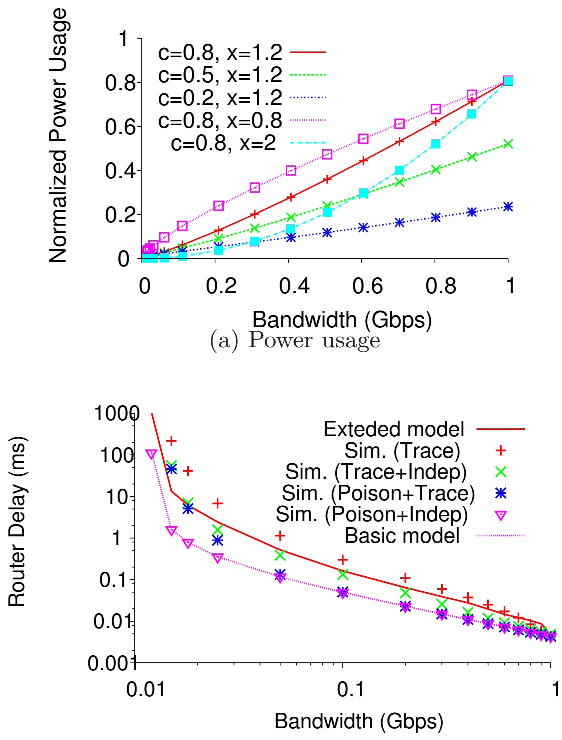


(a) Edge, incoming (b) Core, dir-A

Figure 5: Normalized bandwidth usage. All bandwidths are normalized relative to the peak bandwidth usage during day.

Table 4: Normalized power usage with packet traces.

Scenario		Current $c = 0.8$			Future $c = 0.2$		
	Trace	EEE	eBond	e^{3}B	EEE	eBond	e^{3}B
Two	Edge, in	0.81	**0.16**	0.43	0.22	0.16	**0.13**
	Edge, out	0.81	**0.38**	0.74	0.22	0.38	**0.20**
	Core, dirA	0.81	0.15	**0.12**	0.21	0.15	**0.04**
	Core, dirB	0.82	0.15	**0.12**	0.24	0.15	**0.04**



(a) Power usage

(b) Waiting times

Figure 6: Comparison of model and trace-driven simulations.

tween 14:00-14:15 was used as input to the analytic model. Figure 6(a) shows that the power usage obtained using simulations (markers) and the values obtained using our analytic model (lines) provides a very good match for all example configurations (unique c and x value combination). To help understand where the errors in the waiting times (Figure 6(b)) come from we also include simulation points where we have modified the traces to introduce packet size independence (Trace+Indep), Poisson arrivals (Poisson+Trace), and both (Poisson+Indep).

6. RELATED WORK

Since the initial proposals of Adaptive Link Rate (ALR) technologies for wired networks, many protocols have been proposed [1–3]. This includes both sleep-based energy-aware traffic engineering techniques [17–20] that temporarily put interfaces to sleep within the core network, and a combination of rate switching and active/idle toggling techniques that save energy at the edge [2, 21–23].

Trace-driven simulations [2, 24] and hardware prototypes [25] have been used to study the tradeoff between switching times and energy consumption. Much attention has been given to the EEE standard [3, 6, 26]. This includes the proposal and evaluation of packet coalescaling techniques [3, 27] that increase the burstiness on the outgoing interfaces to improve the energy savings when using EEE. Trace-driven simulations have also been used to evaluate the impact that ALR techniques have on both neighboring routers [28] and end-to-end performance [29]. In contrast, we develop a queue-based model and use it to provide insights to the tradeoffs seen by four general protocol classes.

A few independent analytic models of the EEE protocol have been developed [30–33]. Although there have been some efforts to capture general inter-arrival distributions (e.g., [30]), the majority of these works, similar to ours, assume Poisson packet inter-arrival times. Poisson arrivals have also been shown to provide a good approximation over shorter time scales [9]. Our general router model presented in Section 3.1, which does not make any assumptions about packet inter-arrival times, is inspired by Hohn et al. [8]. Similar to James and Carlsson [28] we extend this model to capture the on-off pattern and energy-tradeoffs associated with EEE. In contrast to the above works, we develop a unifying model that allows us to capture the delay-energy tradeoffs and rate-switching points of both eBond [5] and eeeBond using closed-form expressions.

Finally, it should be noted that similar power saving strategies have been proposed and analyzed in many other contexts, including datacenters [34], datacenter networks [35], individual devices [36], and the wireless interfaces of mobile devices [37]. Also in these contexts hybrid approaches may be beneficial. For example, a datacenter may adjust the number of active servers through dynamic on-off switching [34], and then use speed scaling [38] to adjust the power usage of individual machines at a finer granularity.

7. CONCLUSIONS AND DISCUSSION

This paper presented a generalized protocol evaluation framework in which we perform protocol optimization and performance comparison of four general protocol classes. We first presented a general protocol and modeling framework that captures the energy-delay tradeoffs associated with two orthogonal protocol classes, which uses the on-off toggling of EEE [3, 6] and the interface switching of eBond [5], respectively, as well as a hybrid protocol class (eeeBond) that combines benefits of both protocols. Under Poisson assumptions, we then derived closed-form expressions of the energy-delay tradeoffs of each interface and of the optimal threshold of the packet arrival rates (and link utilizations) that determine the best interface for eBond and eeeBond to use. Finally, we characterized the energy savings possible with the different protocols using both our analytic model and trace-driven simulations. Our results show that eBond typically outperforms EEE, and that it even can outperform eeeBond when interfaces only offer small energy savings when in short-term sleep states (used by EEE and eeeBond). When substantial energy savings in short-term sleep states are possible, eeeBond is by far the best protocol. Interestingly, these findings and results suggest that also when sleep states would allow energy usage of high-power interfaces to become minimal in sleep state, and EEE would become close to energy proportional, the scaling factor x (typically greater than one) of peak energy usage of interfaces with different capabilities is expected to allow eeeBond to significantly outperform EEE by leveraging some slack in the maximum waiting times W^* and lower power-usage of low-power interfaces. While part of this slack also is leveraged with coalescaling techniques [3, 27], these techniques do not leverage the non-proportional advantages of low-power interfaces (and the scaling factor x). Future work includes the development of adaptive algorithms that generalize eeeBond to also turn-off its on-off toggling during times when eBond would otherwise outperform it, allowing us to minimize energy usage at all times of the day.

Acknowledgements

Financial support for this work was provided by CENIIT. The author would like thank Cyriac James for extracting the datasets and for initial discussions regarding the protocols. The author is also grateful to Sara Devenney for help with naming the protocol.

8. REFERENCES

[1] M. Gupta and S. Singh, "Greening of the Internet," in *Proc. ACM SIGCOMM*, 2003.

[2] C. Gunaratne, K. Christensen, , B. Nordman, and S. Suen, "Reducing the energy consumption of Ethernet with an adaptive link rate (ALR)," *IEEE Trans. on Comp.*, vol. 57, Apr. 2008.

[3] K. Christensen, P. Reviriego, M. Nordman, B. andBennett, M. Mostowfi, and J. Maestro, "IEEE 802.3az: the road to energy efficient Ethernet," *IEEE Comm. Magazine*, vol. 48, no. 11, pp. 50–56, 2010.

[4] L. Barroso and U. Holze, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, April 2007.

[5] M. Hähnel, B. Döbel, M. Völp, and H. Härtig, "eBond: energy saving in heterogeneous R.A.I.N." in *Proc. e-Energy*, May 2013.

[6] "IEEE 802.3az energy efficient Ethernet: Build greener networks)," *White Paper from Cisco and Intel*, 2011.

[7] Y. S. Hanay, W. Li, R. Tessier, and T. Wolf, "Saving energy and improving TCP throughput with rate adaptation in Ethernet," in *Proc. IEEE ICC*, 2012.

[8] N. Hohn, K. Papagiannaki, and D. Veitch, "Capturing router congestion and delay," *IEEE/ACM Trans. on Networking*, vol. 17, no. 3, pp. 789–802, June 2009.

[9] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary Poisson view of Internet traffic," in *Proc. IEEE INFOCOM*, Mar. 2004.

[10] N. Tian and Z. G. Zhang, *Vacation Queueing Models: Theory and Applications*. Springer, 2006.

[11] H. Levy and L. Kleinrock, "A queue with starter and a queue with vacations: Delay analysis by decomposition," *Operations Research*, vol. 34, no. 3, pp. 426–436, Jun. 1986.

[12] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation. Vacation and Priority Systems (part 1)*. North-Holland, 1991.

[13] G. Choudhury, "An MX/G/1 queueing system with a setup period and a vacation period," *Queueing Systems*, vol. 36, pp. 23–38, 2000.

[14] R. W. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, pp. 223–231, 1982.

[15] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," in *Proc. USENIX ATC*, 2000.

[16] J. G. Cleary, "Wand project at university of waikato, nz," in *Proc. HPN: Measurements and Analysis Collaborations Workshop*, 1999.

[17] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Proc. IEEE ICC*, 2009.

[18] J. Restrepo, C. Gruber, and C. Machoca, "Energy profile aware routing," in *Proc. IEEE GreenComm*, 2009.

[19] R. Tucker, J. Baliga, R. Ayre, K. Hinton, and W. Sorin, "Energy Consumption in IP Networks," in *Proc. ECOC*, 2008.

[20] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study," in *Proc. IEEE GreenComm*, 2009.

[21] C. Gunaratne, K. Christensen, and B. Nordman, "Managing energy consumption costs in dektop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed," *Int. J. of Netw. Management*, vol. 15, Sept. 2005.

[22] M. Gupta and S. Singh, "Dynamic Ethernet link shutdown for energy conservation on Ethernet links," in *Proc. IEEE ICC*, 2007.

[23] G. Ananthanarayanan and R. H. Katz, "Greening the switch," in *Proc. OSDI*, 2008.

[24] M. Gupta, S. Grover, and S. Singh, "A feasibility study for power management in LAN switches," in *Proc. IEEE ICNP*, 2004.

[25] B. Zhang, K. Sabhanatarajan, A. Gordon-Ross, and A. George, "Real-time performance analysis of adaptive link rate," in *Proc. IEEE LCN*, 2008.

[26] R. Hays, "Active/idle toggling with low-power idle," *Presentation for IEEE 802.3az Task Force*, Jan 2008.

[27] M. Mostowfi and K. Christensen, "Saving energy in LAN switches: New methods of packet coalescing for energy efficient Ethernet," in *Proc. IGCC*, July 2011.

[28] C. James and N. Carlsson, "Green domino incentives: Impact of energy-aware adaptive link rate policies in routers," in *Proc. ACM ICPE*, 2015.

[29] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proc. NSDI*, 2008.

[30] S. Herreria-Alonso, M. Rodriguez-Perez, M. Fernandez-Veiga, and C. Lopez-Garcia, "A GI/G/1 model for 10 Gb/s energy efficient Ethernet links," *IEEE Trans. on Comm.*, vol. 60, Nov. 2012.

[31] M. Marsan, A. Anta, V. Mancuso, B. Rengarajan, P. Vasallo, and G. Rizzo, "A simple analytical model for energy efficient Ethernet," *IEEE Comm. Letters*, vol. 15, pp. 773–775, July 2011.

[32] D. Larrabeiti, P. Reviriego, J. A. Hernandez, J. A. Maestro, and M. Uruena, "Towards an energy efficient 10 Gb/s optical Ethernet: Performance analysis and viability," *Optical Switching and Networking*, vol. 8, pp. 131–138, 2011.

[33] R. Bolla, R.Bruschi, A.Carrega, and F. Davoli, "Green network technologies and the art of trading-off," in *Proc. IEEE INFOCOM Workshops*, 2011.

[34] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, pp. 1378–1391, 2013.

[35] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving energy in data center networks," in *Proc. NSDI*, San Jose, CA, 2010.

[36] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: Eliminating server idle power," in *Proc. ASPLOS*, 2009.

[37] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," in *Proc. MobiSys*, 2010.

[38] A. Wierman, L. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," pp. 2007–2015, 2009.