Green Domino Incentives: Impact of Energy-aware Adaptive Link Rate Policies in Routers

Cyriac James
University of Calgary, Canada
cyriac.james@ucalgary.ca

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

ABSTRACT

To reduce energy consumption of lightly loaded routers, operators are increasingly incentivized to use Adaptive Link Rate (ALR) policies and techniques. These techniques typically save energy by adapting link service rates or by identifying opportune times to put interfaces into low-power sleep/idle modes. In this paper, we present a trace-based analysis of the impact that a router implementing these techniques has on the neighboring routers. We show that policies adapting the service rate at larger time scales, either by changing the service rate of the link interface itself or by changing which redundant heterogeneous link is active, typically have large positive effects on neighboring routers, with the downstream routers being able to achieve up-to 30% additional energy savings due to the upstream routers implementing ALR policies. Policies that save energy by temporarily placing the interface in a low-power sleep/idle mode, typically has smaller, but positive, impact on neighboring routers. Best are hybrid policies that use a combination of these two techniques. The hybrid policies consistently achieve the biggest energy savings, and have positive cascading effects on surrounding routers. Our results show that implementation of ALR policies can contribute to large-scale positive domino incentive effects, as they further increase the potential energy savings seen by those neighboring routers that consider implementing ALR techniques, while satisfying performance guarantees on the routers themselves.

Categories and Subject Descriptors

C.2.0 [**Computer-communication Networks**]: General—*Data communications*; C.2.6 [**Computer-communication Networks**]: Internetworking—*Routers*

Keywords

Energy Efficiency; Adaptive Link Rate; Energy Proportional Computing; Router Performance

ICPE'15, Jan. 31–Feb. 4, 2015, Austin, Texas, USA.

http://dx.doi.org/10.1145/2668930.2688045.

1. INTRODUCTION

Internet routers are typically over provisioned and operate at low utilization, leaving much room for energy savings. Motivated by increasing energy prices and high CO_2 emissions (associated with non-green energy, for example), different Adaptive Link Rate (ALR) policies and techniques [10, 14, 16] have been proposed to reduce energy consumption when routers are lightly loaded.

Depending on hardware capability, ALR policies can operate at different time scales. Over larger time scales (e.g., order of minutes), the operator can save energy by adapting the active service rate of interfaces based on the estimated utilization. At the granularity of tens of microseconds, a router can save energy by toggling between an active high-power mode and a low-power idle mode, during which some interface components are put to temporary sleep.[1] This finer granularity allows decisions to be made based on the current packet arrival pattern and queue occupancy.

In general, ALR policies attempt to scale the energy usage based on the current traffic load. Ideally, routers should be energy proportional [3]. In this case, idle router interfaces do not consume any power and the energy consumption is proportional to the load. While policies that use low-power idle modes can be implemented within the recent Energy Efficient Ethernet (EEE) standard [10] and the feature is already available in the market (e.g., Cisco Catalyst 4500E switches), achieving proportional energy usage without significant delay penalties is typically not possible with the current state of the art hardware. It should also be noted that techniques have been proposed to allow "near" proportional energy consumption using non-proportional hardware available today. For example, eBond [18] uses redundant heterogeneous links coupled with energy-aware bonding to save energy. With this approach, a low-bandwidth link is used when the router is lightly loaded, allowing the regular high-bandwidth link to be turned off. With hardware expected to become increasingly energy proportional, it is therefore important to consider the impact of implementing ALR policies on both proportional and nonproportional systems.

Although many ALR policies have been proposed and evaluated, not much is known about the effect that a router applying ALR techniques has on the performance and po-

[1]Throughout the paper we will use *low-power idle mode* and *sleep mode* interchangeably. We will also use a relatively broad definition of ALR policies, which include both policies that adapt the rate itself and policies that enter such low-power mode, for which the service rate is zero.

tential energy savings of neighboring routers. As these techniques are being increasingly deployed, it is important to understand the impact such deployment may have on the overall end-to-end system. For example, are there global performance or energy penalties associated with routers greedily minimizing their own energy usage? And, perhaps more importantly, does the potential energy savings on other routers increase or decrease with the implementation on one router?

In this paper, we use trace-driven simulations to analyze the effects that ALR policies used on one router have on neighboring routers. We first develop a simple evaluation framework, that allows us to evaluate different classes of ALR policies. Our framework captures the basic tradeoffs between *energy usage* and *per-router packet delay* for each of the policy classes. Using trace-based simulations, we then evaluate policies under a wide range of traffic patterns, and provide insights with regards to the energy-delay tradeoff effects these policies have on neighboring routers.

The model developed for our evaluation framework captures the energy and delay characteristics of two general ALR policy classes, and hybrids thereof. The first policy class, *rate switching policies*, saves energy by adapting the (maximum) service rate of the outgoing interfaces. Referring to the above discussion, we note that rate switching can be implemented either by adapting the service rate of a single interface [14] or by implementing heterogeneous bonding [18]. The second policy class, *active/idle toggling policies*, saves energy by temporarily placing the interface into a low-power idle mode when there are no (or few) packets to process. Finally, the hybrid policy class adjusts the active service rate based on long-term measurements and also use active/idle toggling to save energy at shorter time scales.

Motivated by an end-to-end client-server scenario, we use both edge and core network traffic traces, and consider simulation scenarios in which (i) the traffic is aggregated at routers with increasingly higher capacities, (ii) the traffic is dispersed on its way towards the edge of the network, and (iii) cases with varying degrees of traffic multiplexing. Particular attention is given to the average energy savings and the tail of the per-router packet delays, as values such as the 99^{th} percentile often are important in practice.

Our results provide a quantitative comparison of the relative impact different policies have on neighboring routers under different workload scenarios and traffic patterns. We find significant differences in the possible energy savings at neighboring routers. The savings are largest on upstream interfaces close to the edge, which typically carry a larger fraction small packets, but reduce with increased multiplexing of packet streams. Perhaps most interestingly, for all three policy classes, we observe that implementing ALR policies in upstream routers allows downstream routers to achieve higher energy savings than is possible if the upstream routers do not use ALR. While the additional improvements in energy savings is greatest for rate switching, which can achieve up-to 30% additional energy savings, the other two policy classes can also achieve up-to 5-10% additional energy savings.

These results suggest that greedy energy savings on one router can have multiplicative benefits. First, they reduce the energy usage on the router itself. Second, they increase the potential energy savings possible on neighboring routers, further incentivizing implementation of ALR techniques. With the largest energy savings and significant energy savings improvements, our findings make a strong case for the more advanced hybrid policies, when possible. Of course, it is important to also note that the basic rate switching and active/idle toggling policies can provide significant advantages on their own, and should hence not be ignored. Particularly as hardware constraints and availability of utilization and packet-level queueing information may differ between routers and operators.

The remainder of this paper is organized as follows. Section 2 sets the context, describes the state of the art, and the policies analyzed in the paper. Section 3 presents the system model and datasets used to evaluate the impact of implementing different policies. Section 4 presents our performance and energy implication analysis. Finally, Section 5 concludes the paper with a discussion of our findings.

2. BACKGROUND AND RELATED WORK

2.1 End-to-End Path and Energy Usage

HTTP is responsible for a majority of the Internet traffic [24]. To understand the overall energy usage of present day networks and the impact one router's actions have on that of the next router along the path, we first consider a Web request being sent between a home user and a server in a modern datacenter. Today, the big players are (i) building big data centers in places where electricity and network bandwidth are cheap, and/or (ii) moving the content closer to the end user by using CDNs or putting their own servers in the datacenters operated by the ISPs. This means that some traffic will be served close to the end user, while other traffic will need to traverse the entire end-to-end path. Such end-user generated traffic is aggregated as it moves into the core networks and then disperse again as it moves closer to the particular datacenter serving the request. Naturally, the HTTP response takes a similar but reverse path.

Along its path, a typical Internet packet traverse many routers, operated by different operators or Autonomous Systems (AS), each with its separate administrative domain and policies. For example, a simple IP-to-AS mapping [26] analysis that we performed on 1,000 randomly selected traceroutes from the Route Views project[2] suggests that an average packet may see 13.1 routers and 4.2 ASes. With many operators along the path, the choices made by one operator will clearly impact others. Even within an AS, the choices made on one router will impact neighboring routers.

Taking a birds-eye view, edge networks consume 70-80% of the total network energy, and core networks the remaining 20-30% [4], with the difference explained by the edge being responsible for 95% of the network elements [4,5]. However, recent studies [25] combined with a doubling in traffic volumes every 18 months [37], suggest that the energy shares will be comparable around year 2021. These numbers show that it is important to consider energy saving implications on both core and edge network routers.

2.2 Adaptive Link Rate Policies

In 2003, Gupta and Singh [16] first discussed Adaptive Link Rate (ALR) as a plausible energy efficient solution for wired networks. Building on technologies such as Dynamic Voltage Scaling [34], they argue that apart from implement-

[2] U. of Oregon, Route Views project. `http://www.routeviews.org/`

ing these techniques in the line card, the main challenges may be determining (i) when to change the link rates and (ii) what are the performance implications on the network. Since then, many ALR techniques and policies have been proposed that address (i) by adapting the link rate based on traffic measures such as queue sizes, link utilization, or a combination thereof [10,14,16]. However, not much is known about (ii) and the impact that a router implementing ALR techniques may have on neighboring routers. Addressing this question is our primary contribution.

In this paper, we consider two general classes of ALR policies, and a hybrid thereof.

- **Rate switching:** With rate switching, the active link rate μ is varied (linearly, stepwise, or by any other function) depending on the traffic load, with rate changes typically happening over time scales of minutes. We consider a general policy class, but note that the service rate changes can be implemented in many ways, including solutions using a single interface or heterogeneous bonding [18].

- **Active/idle toggling:** With active/idle toggling policies, the interface operates on a much finer time granularity and frequently toggle between a low-power idle mode and a high-power active mode. A router interface switch to low-power idle mode when there are no packets to serve, and switch back to active mode when L bytes have been queued. Typically, there will be a time delay Δ before the interface is activated and can start serving the queued packets.

- **Hybrid:** A general hybrid policy adjusts the active service rate μ on long-term basis (e.g., order of minutes), and uses active/idle toggling with a threshold L to save energy at shorter time scales. To restrict ourselves to a single protocol parameter, we consider an example policy with a very small threshold $L = \epsilon$, such that the interface always is activated when a new packet arrives. A discussion on the impact of this parameter choice is provided in Section 4.2.

2.3 Energy Savings

Implementations based on the first two general policy classes above have been evaluated for a wide range of systems. Most papers that discuss ALR techniques at the core network have focused on the use of sleep-based energy-aware traffic engineering techniques [7, 8, 30, 33] that allow some interfaces to go to sleep temporarily. On the other hand, end hosts, access networks, and edge technologies have been evaluated under both rate switching and active/idle toggling techniques [2, 13, 14, 17]. Trace-driven simulations [14, 15] and hardware prototypes [36] have been used to study the impact of switching times and policies on the energy consumption when implementing ALR techniques in the Ethernet. It has also been shown that finer time granularity is needed for bursty traffic, such as Internet traffic [27, 35]. Other works have considered the impact on higher layer protocols such as TCP [19].

Despite this body of work, there is very limited work studying the impact of ALR techniques on neighboring routers. This question is particularly important as routers do not operate in isolation and implementation of these "green" techniques on one or more routers will impact the overall network performance, as measured by the packet delivery delays, for example, as well as the potential energy savings others may be able to achieve without sacrificing performance.

Perhaps most closely related is the work by Nedevschi et al. [28], which simulate the end-to-end performance (measured in terms of end-to-end packet delay and loss) of the network as a whole, when ALR techniques are applied to intermediate routers/switches along the end-to-end path. In contrast, we study the impact ALR techniques have on neighboring routers, allowing us to provide insights into how the use of ALR techniques may affect neighboring routers' potential benefits of using ALR and their decision to use ALR techniques. A broader class of policies also allow us to capture differences and similarities across policy classes.

2.4 Standardization and State of the Art

The recent Energy Efficient Ethernet (EEE) standard [1, 10], called *IEEE 802.3az*, is based on an active/idle toggling framework by Hays [20], and defines the signaling that is required between the transmitter and the receiver when the former toggles back-and-forth between a Lower Power Idle (LPI) mode and an active mode. The standard does not define policies to determine when to change mode, but it has been suggested that the default wakeup time typically should be approximately equal to the transmission time of the maximum length packet in the particular link [1]. For example, for a 1 Gbps link, it takes ≈ 0.01ms to send a 1500 byte packet. Based on simulations done in lab environment, the expected power saving for IEEE 802.3az enabled Cisco Catalyst 4500E, a 384 1000Base-T port switch, is 74% [1]. It should also be noted that standard Gigabit Ethernet interfaces already support multiple data rates (e.g., 10 Mbps, 100 Mbps and 1 Gbps) using the auto-negotiation feature, which can be utilized for implementing *rate switching*.

3. SYSTEM MODEL AND DATASETS

3.1 Basic Router Model

For the purpose of our trace-driven evaluation we use the router model developed by Hohn et al. [21]. The model assumes a First-In-First-Out (FIFO) queueing policy, was developed and validated using real traffic, and ratifies the commonly held assumption that the output buffer is the bottleneck in popular store-and-forward routers that implements virtual output queueing (to avoid head-of-line blocking). Typically, the switch fabric is overprovisioned and very little queueing happens at the incoming interfaces. With very small switch fabric delays (typically $10 - 50$ μs) and by focusing on the tail of the delay distribution, for which acceptable delay constraints may be of the order of milliseconds or above, the model only considers the queueing delay on the outgoing interface and the transmission delay. Finally, motivated by the low loss context and that line cards often can accommodate up to 500 ms worth of traffic, the model assumes an infinite buffer size.[3] Under these assumptions, the delay d_k experienced by the k^{th} packet in an *always-on* router with service rate μ is

$$d_k = [d_{k-1} - (t_k - t_{k-1})]^+ + \frac{l_k}{\mu}, \qquad (1)$$

[3]For our experiments, we are typically interested in 99^{th}-percentile per-router delays below 10^2 ms.

where $[y]^+ = \max(y, 0)$, and t_k is the arrival time of the k^{th} packet of length l_k. For additional details on the model, we refer the interested reader to the original paper [21].

3.2 Policy Model

We next extend the router model to capture the performance under the three general ALR policies, defined in Section 2.2. Assuming that the service rate does not change during service of a packet, rate switching, between non-zero service rates, is easily modeled by giving each packet k an individual service rate μ_k. To capture the active/idle toggling aspect we introduce two additional parameters Δ and L, where L is the pre-defined threshold parameter used by the active/idle policy and Δ is the time that it takes to activate the link again after being in the low-power idle mode.

Given these assumptions, a general hybrid policy that allows active/idle toggling should (i) switch to low-power idle mode when there are no packets to serve, and (ii) switch back to active mode when L bytes have been queued. The delays of such a policy can be modeled as follows:

$$d_k = \begin{cases} \Delta + W_k + \sum_{n=k'}^{k} \frac{l_n}{\mu_n}, & \text{if } t_k \text{ during idle} \\ [d_{k-1} - (t_k - t_{k-1})]^+ + \frac{l_k}{\mu_k}, & \text{otherwise,} \end{cases} \quad (2)$$

where W_k is the waiting time until the interface goes active after a packet k arrive at the interface when it is in low-power idle mode, calculated as $W_k = (t_{k''} - t_k)$, k' is the first (lead) packet that arrives during an idle period, and k'' is the packet that takes the interface out of low-power idle mode. We note that the lead packet during an idle period satisfies the condition $t_{k'} > t_{k'-1} + d_{k'-1}$ and that the packet that takes the interface out of low-power idle mode can be calculated as $k'' = \min_{k''}\{k'' | k'' \geq k' \cap \sum_{n=k'}^{k''} l_n \geq L\}$.

As previously argued, we only consider policies for which the *active* service rate μ changes on the order of minutes and therefore (at finer time granularity) use a constant service rate $\mu_k = \mu$. For our active/idle policy, we always use the full link rate $\mu = \mu^{max}$, but simulate the policy for different thresholds L. For the hybrid policy, we pick a threshold $L = \epsilon$ smaller than the size of the smallest packet, and simulate the policy with different active service rates $\mu \leq \mu^{max}$.

3.3 Energy Model

Motivated by advancements in Dynamic Voltage Scaling and measurements of existing routers [18], we model the power usage $P_a(\mu)$ of a link with active link rate μ, using a simple linear model between the minimum and maximum service rate:

$$P_a(\mu) = P_a^{min} + (P_a^{max} - P_a^{min}) \frac{\mu - \mu^{min}}{\mu^{max} - \mu^{min}}, \quad (3)$$

where P_a^{max} is the maximum active energy usage per time unit (power) when the router interface operates at maximum service rate μ^{max}, and P_a^{min} is the power usage when the interface operate at lowest possible service rate μ^{min}. In general, this power is lower bounded by the power usage P_s in sleep (low-power idle) mode; i.e., $P_s \leq P_a^{min}$.

Ideally, future routers will be energy proportional [3]. For this to be the case the energy usage should be proportional to the computing/service provided. This would imply that (i) the power usage P_a when in active mode is proportional to the service rate μ, (ii) the power usage P_s when in sleep mode is 0, and (iii) the activation time Δ and activation energy is 0.

Unfortunately, routers are not yet fully energy proportional. First, the interfaces often consume a significant amount of power (w.r.t to the maximum power) when it is not processing any traffic or when in various sleep modes. In some cases, an interface even consume a significant amount of power when it is down or when the cable is unplugged [18]. Second, there are non-negligible time delays and energy costs associated with activating an interface in the case of routers with active/idle toggling features [1, 31]. In the short term, very few circuits in the physical layer can be turned off (put to sleep) during the low-power idle mode, resulting in modest energy savings.[4] Hence, we make a pessimistic assumption that $P_s = P_a^{min}$, by raising the value of P_s, and $\mu^{min} = 0$. (We note that this assumption also is valid for the best-case scenario in which routers are energy proportional and $P_s = P_a^{min} = 0$.) Furthermore, we assume that the activation time Δ is constant and that the activation energy is proportional to the power usage $P_a(\mu)$ in active mode and the activation time Δ. Under these assumptions, the total energy usage is equal to

$$T_a P_a(\mu) + T_s P_s + c N_a \Delta P_a(\mu), \quad (4)$$

where T_a is the total time in active mode, T_s is the total time in sleep mode, N_a is the number of times the interface have been activated, and c is a constant capturing the energy penalty associated with turning on the interface. When $c = 0$ there is no energy cost associated with the activation time, and when $c = 1$ the energy usage is the same as when active.

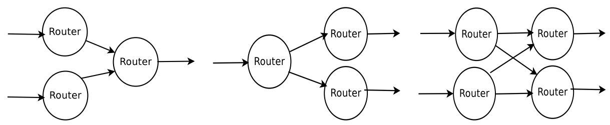
3.4 Traffic Model

To model the traffic seen at consecutive routers, we use a basic model with two layers of back-to-back routers. Each of the routers in the first layer is assumed to have m_{in} input interfaces, m_{out} output interfaces, and each outgoing interface sees $1/m_{out}$ of the traffic from each of the incoming interfaces. The corresponding number of interfaces for the second set of routers are n_{in} and n_{out}. The packets from the output interfaces of the first set of routers become input sequence of packets to the second set of routers.

For each interface, we use a separate packet trace. To decide which packets to forward to each output interface, we leverage the destination addresses in the Internet Protocol (IP) headers. More specifically, we identify m_{out} (or n_{out}) blocks of IP addresses, each consisting of many IP prefixes, such that on average each block has the same fraction of total traffic. We then forward the traffic of each separate block to individual interfaces. While the IP addresses in the traces have been anonymized with Crypto-PAN [12], we note that Crypto-PAN is prefix preserving and therefore allow us to capture the longest-prefix-based routing used by typical store-and-forward routers, used on the Internet.

For simplicity, we restrict our analysis to the case of routers with the same number of incoming and outgoing interfaces; i.e., when $m_{in} = m_{out} = m$ and $n_{in} = n_{out} = n$. Under this abstraction, we look at scenarios in which the traffic is increasingly aggregated ($m > n$), is dispersed as it is moving towards the edge ($m < n$), and symmetric scenarios with different degrees of multiplexing (i.e., different values

[4] In the long run, the expectation is that advanced hardware technologies will allow energy savings up to 80% [1].



(a) Aggregation (b) Dispersion (c) Multiplexing

Figure 1: Modified tandem queue model.

of $m = n$). Figure 1 provides an overview of three basic example scenarios: 2×1, 1×2, and 2×2.

We note that most routers have few interfaces, suggesting that m and n typically are small. Furthermore, as the load across interfaces typically is highly skewed, with most of the traffic being directed to a small subset of the interfaces, the insights of small m and n may be applicable even for routers with many interfaces.

3.5 Packet Traces

Simulating an $m \times n$ topology, we use mn packet traces. Each trace is fed into a separate incoming interface of the m first-layer routers, each relaying $1/n^{\text{th}}$ of their traffic to each of the n second-layer routers.

For the core, we use packet traces collected at a core router connected to a trans-pacific link, labeled *samplepoint-F*, in the WIDE Internet (MAWI) dataset[5] [9]. The traces are collected between 14:00 and 14:15 (local time) each day of January 2013. For the edge, we use traces from the Waikato Internet Traffic Storage (WITS) project[6] [11], labeled Waikato VIII, and are collected from an edge router of a university network. For simplicity, we use the same 15 minute time-of-day period as used for the core. Appendix A provides a more detailed characterization of the traces.

4. PERFORMANCE ANALYSIS

4.1 Methodology overview

This section presents our simulation results. Using the packet traces and simulation methodology described in Section 3.5, both the delay (Section 3.2) and the energy usage (Section 3.3) associated with each policy are measured over the simulation duration. Throughout our analysis we use an initial warm-up period, and do not include the initial packets in our statistics. Unless stated otherwise, we conservatively use $c = 1$ in our evaluation.

Both the energy usage and the per-packet routing delay are measured variables, which depend on the traffic pattern and the protocol parameters used by each protocol. To capture the general performance tradeoff between these variables, we run a sequence of simulations with the same workload, but in which we vary the main system parameters associated with each policy. In the case of the rate switching policy and the hybrid policy, different energy-delay tradeoffs are achieved by running simulations with different active service rate μ (and hence also $P_a(\mu)$). In contrast, the energy-delay tradeoff seen by the active/idle toggling policy is determined by the byte threshold L. To illustrate these tradeoffs, Figure 2 shows the 99-percentile per-router packet delay, as a function of the corresponding protocol parameter. The results are for the outgoing edge traffic (Section 3.5), and as per the EEE standard specifications, in all cases, we use Δ equal to the packet processing time (≈ 0.01 ms) of the largest possible packet when operating at maximum link capacity μ^{max}.

Furthermore, in later sections, where we show results for a particular delay (e.g., the energy saving or improvement in energy savings as a function of the delay; Figures 6, 7, 8 and 9) we must perform a binary search over the primary policy parameter (link rate or threshold value, that control the delay-energy tradeoff) for the achieved delay. By identifying the delay-energy pair (both measured variables) when the two scenarios or policies see the same delay (but different energy usage or energy savings), that provide a fair head-to-head comparison for a given target delay. In these cases, we call the "per-router packet delay" seen by a packet (at a single router) the "target per-router packet delay".

4.2 Single Router Energy Analysis

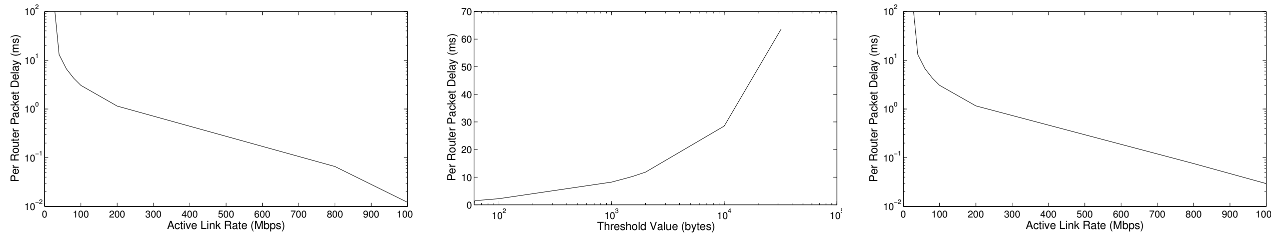
Before our analysis of back-to-back routers, we first present results for a single router implementing each of the three basic policy classes, defined and modeled in Sections 2.2 and 3.2, respectively. For clarity, we only present example results illustrating the relative performance and/or energy savings with each policy. Furthermore, for the single-router case presented in this section, we use $m = n = 1$.

Figure 3 shows a head-to-head comparison of the normalized energy usage of the different policies, as a function of the 99-percentile per-router packet delay. Normalization is done with regards to the regular energy usage when operating at maximum power P_a^{max}. Results are presented for 15 minute example traces on the core (dir-A) and edge router (outgoing). The "proportional" cases assume $P_s = P_a^{min} = 0$. (Here, $c = 0$ for both the active/idle toggling and hybrid policy.) This case is motivated by future hardware improvements, as well as software solutions achieving energy proportionality with non-proportional hardware. For the "conservative" cases, we use $c = 1$ and some of the most pessimistic parameter values that we observed in the profiling literature [1, 18, 19, 31]; all values reported by Hähnel et al [18]. For the 1 Gbps edge router we used $P_s = P_a^{min} = 1.35$ W and $P_a^{max} = 1.92$ W. For the 10 Gbps core router, the corresponding values are 7.88 W and 8.10 W, respectively.

When implementing rate switching and hybrid policies on non-proportional hardware, these policies typically can only select rates from a pre-defined set of link rates. While we show results for the full range of delay values, in practice, the tradeoff curves for these policies are therefore expected to be stepwise. As such, the presented results only illustrate approximate energy-delay tradeoffs.

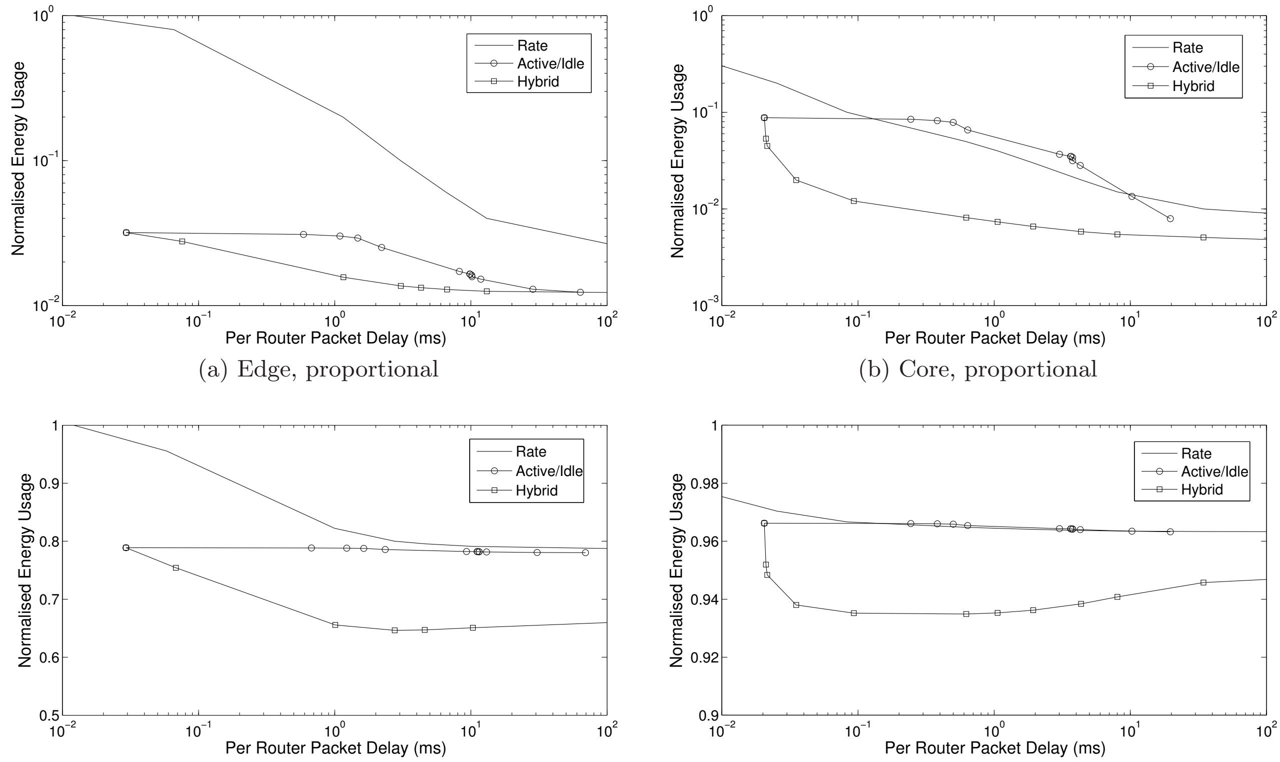
Typical systems likely would see savings in-between the "proportional" (Figures 3(a) and 3(b)) and "conservative" (Figures 3(c) and 3(d)) extremes for a foreseeable future. While the big difference between the possible energy savings using energy proportional hardware and the conservative hardware specs may appear disheartening at first, it is important to remember that systems such as eBound [18], can achieve energy proportional savings using non-proportional hardware. In fact, we argue that eBond [18] could easily be extended to use active/idle toggling on each link, allowing also the hybrid policy to be implemented with current hardware. The proportional scenarios can provide insights on the potential performance of such systems.

[5]MAWI, http://mawi.wide.ad.jp/mawi/, June 2013.
[6]WITS, http://www.wand.net.nz/wits/, June 2013.



(a) Rate switching (b) Active/idle toggling (c) Hybrid

Figure 2: Impact on the 99-percentile per-router packet delay when varying the main parameter of each policy.



(a) Edge, proportional (b) Core, proportional

(c) Edge, conservative (d) Core, conservative

Figure 3: Normalized energy usage for each policy, under four example scenarios.

We note that the particular hybrid policy used in our simulations, similar to the other policies, is restricted to use a single protocol parameter. With this observation in mind, it is interesting that the biggest energy savings consistently (across scenarios) are achieved by the hybrid policy. While most of the energy savings come from the active/idle part of the policy, these results show that it is better to adjust the active rate μ (at a larger time scale) and turn the interface back on (to active mode) as soon as one receives a single packet, than to use the maximum link rate and adjust the byte threshold L (as done by the active/idle policy).

Note that an optimal hybrid policy that optimizes over both L and μ would do even better. For example, the increase in energy usage for the hybrid policy under high delays is due to additional queueing caused by low link rates. (In the limit, the performance of the hybrid policy would converge to that of the rate switching policy.) For these delays, it would be better to use the rate used for the local minimum on the curve and instead increase the byte threshold L.

4.3 Back-to-Back Rate Switching Example

Applying ALR techniques can affect the per-router packet delays and potential energy savings of neighboring routers. In this section, we use a simple example scenario to illustrate how two back-to-back routers can be affected. Here, rate switching is implemented on one or both of the routers.

Figure 4 shows the CDF of the per-router packet delay seen at each of the two routers, for three example configurations. The first corresponds to the default configuration, in which both interfaces operate at maximum capacity. In the second configuration, the link rate of both routers are decreased by a factor 12.5, and in the third configuration only the link rate of the first is decreased. We note that the tail delays (upper percentiles) are significantly lower on the second router, especially for the cases when the link rates of the first router are reduced. In fact, we often observe a decrease in tail delays on the second router just by lowering the service rate on the first router. This illustrates that rate switching can have positive effects on neighboring routers.

At this point, it should be noted that the reduction in tail delays depend on the traffic patterns observed in our packet traces. In fact, at first, a reduction can appear somewhat contrary to what may be suggested by traditional two-stage tandem queue models [6, 29]. For example, Burke's theorem [6] suggests that two consecutive M/M/m queues with independent service times can be treated independently, and any scaling in the service rates of the first router should not benefit the second router. Furthermore, when the service times are the same at the two routers and the routers are lightly loaded (e.g., $\rho < 0.6$), the second M/M/1 $\rightarrow$ M/1 router would typically see higher delays [29]. This can be explained by queued jobs on the first router typically arriving during service of a very large job, which because of the bigger job size also will be queued at the second router too; this time for an even longer time duration. However, as discussed in Appendix A, in contrast to assumptions common in these studies, for all our traces, the packet size distributions are well approximated by a bimodal distribution (Figure 10(b)), and service times are highly correlated both with regards to back-to-back packets (Table 1) and processing at consecutive routers [21].

When discussing the related tandem-queue literature, it should be noted that both a richer set of service time distributions and correlated service times have been considered (e.g., [29, 32]). However, often these studies use continuous service time distributions and potentially miss effects of the bimodal packet-size distribution and the inter-packet correlations seen in real network traces. For example, Sandmann [32] recently simulated the end-to-end packet delays through a series of queues, with correlated service times drawn from "general" (but continuous) service time distributions. While their results provide interesting insights into the relative impact as the load of the system changes from light-to-heavy load, the simulations also suggest that under light load correlated service times typically result in an increase in the end-to-end delays. In contrast, we typically observe a decrease both under light and heavy load.

To help explain how the above properties can result in a decrease in the delay seen on the second router, consider a simple 1×1 model with two packet sizes: large and small. Furthermore, assume that both routers have the same link capacity. In this special case (i) no queueing of large packets will happen on the second router, and (ii) all small packets queued behind a large packet on the first router will be queued for the difference in processing time of a large and small packet on the second router. These delays corresponds to the rightmost (maximum delay) points of the R_2 curves in Figure 4. Under these circumstances, the first router can see much larger delays, as packets can be queued behind more than one large packet. The somewhat larger median values, are due to small packets arriving to an otherwise empty system during processing of a large packet. These packets see a smaller delay on the first router, but would not greatly affect the average, which is dominated by the tail values. For most of our analysis we focus on the tail.

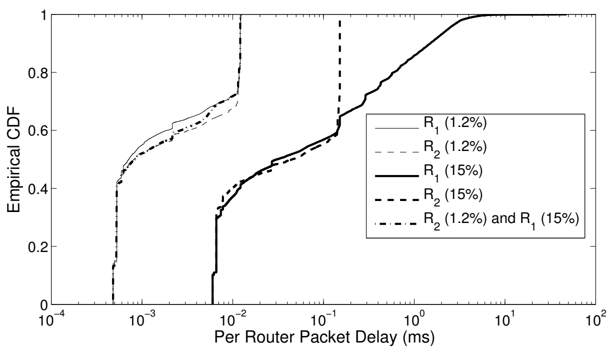


Figure 4: CDF of the per-router delay on two back-to-back routers R_1 (solid lines) and R_2 (dotted lines) when one or both their link rates are adjusted to different alternative link utilizations (shown in label). (Outgoing edge trace.)

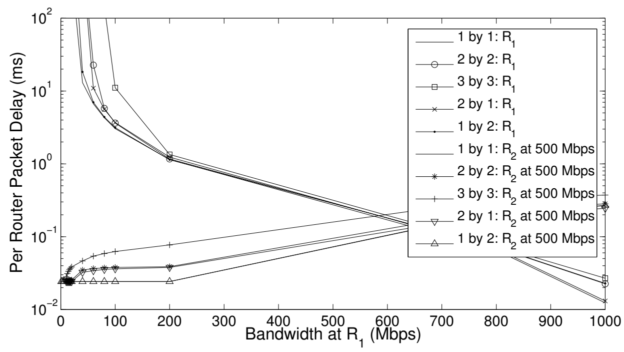


Figure 5: The 99-percentile per-router packet delays for each of the two routers (R_1 and R_2), under different link rates (shown on x-axis for R_1 and in the label for R_2) and scenarios (label).

To look closer at the interplay between the back-to-back routers, we next consider the per-router packet delays under different degrees of multiplexing. Figure 5 shows the 99-percentile delays observed at the two routers (R_1 and R_2) as a function of the link rate of the outgoing interfaces of the first router (x-axis), for different link rates of the outgoing interfaces of the second router. We again note that the more rate constrained (higher link utilization) the first router (R_1) is, the smaller the delays on the second router (R_2) become, at the cost of higher delays on the first router. Furthermore, for most of the cases, the delays on the second router are lower than the delays observed on the first router when the two routers have the same link rate (in this case 500 Mbps). In these cases, the combined per-packet delay (summed over the two routers) is dominated by the per-packet delay on the first router.

Motivated by the above observations, we next consider the potential energy savings when applying rate switching at both routers. Figure 6 shows the energy savings on each of the two routers for different degrees of multiplexing. All energy savings are calculated relative to the case when the router interfaces operate at full link capacity. For this and the remaining analysis, we focus on the proportional case.

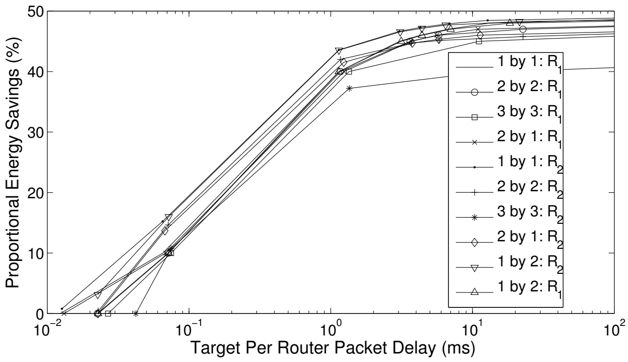


Figure 6: Energy savings for 99-percentile target per-router packet delays, under different scenarios (shown in label).

While there are regions for which the savings are greater on the first router, we note that the delay region for which the energy savings are larger for the second router are substantial. With the exception for the 3x3 case, the savings are always bigger for the second router. For the 3x3 case, there is a significant amount of multiplexing adding randomness at the second router that is not present on the first set of routers (regardless if ALR is used or not). This results in a significant delay penalty, especially under high utilizations.

4.4 Cascading Energy Improvements

The example in the previous section illustrates two types of energy improvements. First, downstream routers often see larger energy savings than upstream routers. Second, and perhaps more interestingly, the downstream routers themselves typically are able to achieve larger energy savings with ALR methods when the upstream routers also implement ALR methods, compared to if the upstream routers do not. This suggest that implementing ALR methods can help further incentivize neighboring routers to implement ALR methods, potentially leading to positive cascading effects.

In this section, we characterize this second type of multiplicative improvements. Under different scenarios and ALR policies, we quantify the improvements in energy savings at router R_2 that can be contributed to implementing ALR techniques at router R_1. We define the improvement in energy savings as the difference between (i) the energy savings that router R_2 can make when router R_1 is implementing the given ALR method, and (ii) the energy savings that router R_2 can make when router R_1 does not implement ALR.

Figure 7 shows the improvements for the edge and core traces when using the rate switching policy. We note that the largest improvements are observed for the outgoing edge traffic (the case with the smallest packets and lowest utilization) when multiplexing is low. For the 1x1 case, additional energy savings of up-to 30% are possible for all traces.

Also for the active/idle toggling policy (Figure 8(a)) and the hybrid policy (Figure 8(b)) we observe significant improvements, although smaller than the peak improvements observed for the rate switching policy. The two sharp peaks in improvements observed for some of the active/idle policy curves correspond to threshold values of approximately the same size as large and small packet, respectively. As exemplified by the delay-shift of the peaks in Figure 9, this effect causes the peaks to be shifted depending on the utilization. Different utilizations cause the peaks to occur for different delay values. Relative to the active/idle policy, we also see that the hybrid policy shows relatively smaller but consistent improvements across workloads.

Finally, we note that while there are cases under low utilizations where ALR techniques can result in reduced energy savings (although still savings), these regions are much smaller and not the regions for which the ALR methods are likely to operate (such as to ensure good energy savings). For example, for intermediate 99-percentile per-router packet delays (between approximately 0.025 ms and 10 ms, for example) the improvements are positive for all policies and workloads considered.

5. DISCUSSION AND CONCLUSION

ALR policies and techniques can provide significant energy savings (e.g., Figure 3), providing strong incentives for operators to implement them into their routers. In this paper, we present a trace-based analysis of the impact that a router implementing these techniques has on neighboring routers. Looking at three general policy classes, each with their own hardware and monitoring constraints, we show that (i) ALR policies of each class have positive impact on the potential energy savings of neighboring routers, and (ii) the absolute energy savings at neighboring routers significantly depends on workload scenario and traffic patterns.

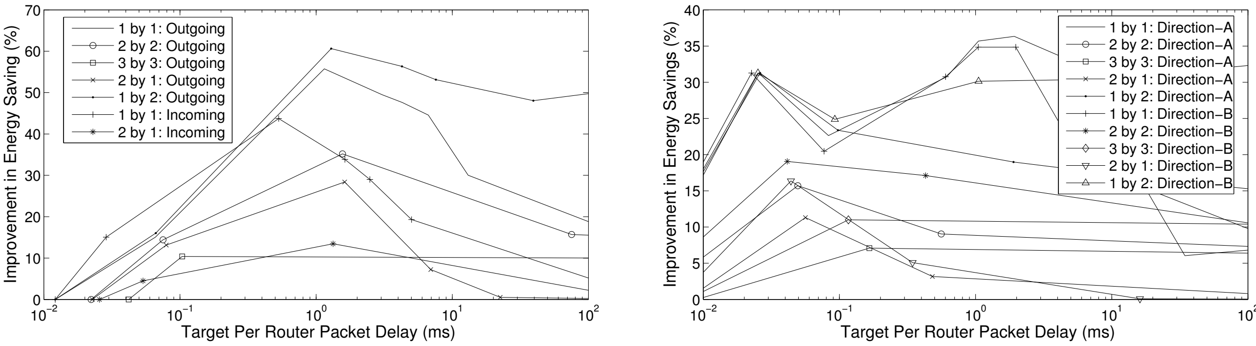
Tying back with our discussion of the end-to-end path of a packet, we note that the biggest savings are achieved at upstream interfaces close to the edge. These streams typically carry a relatively larger fraction small packets (e.g., TCP acks and HTTP requests) and can hence benefit more from ALR policies. However, these savings reduce with increased multiplexing of packet streams.

The biggest improvements in energy savings are achieved with rate switching policies. These policies can result in upto 30% additional energy savings (Figure 7) on the neighboring downstream routers. These results suggest that a greedy energy savings on one router can have green cascading effects and provides further incentive to implement these policies at a large scale.

While hardware that allows active/idle toggling can achieve great energy savings (Figure 3), the multiplicative effects of these policies are somewhat smaller, although still positive (Figure 8(a)). Perhaps most attractive are the hybrid policy, which achieves the largest energy savings (Figure 3), and have positive effects on neighboring routers (Figure 8(b)). In the absence of energy-proportional hardware, we envision that effective hybrid policies could be implemented by combination of heterogeneous bonding [18], with active/idle toggling (based on the EEE standard [10]) implemented on each redundant link. Future work will consider the implications of large-scale deployment and the interaction with higher layer protocols [22]. More complex router models and an investigation of the variability (beyond our 99-percentile analysis) also present promising directions for future work.

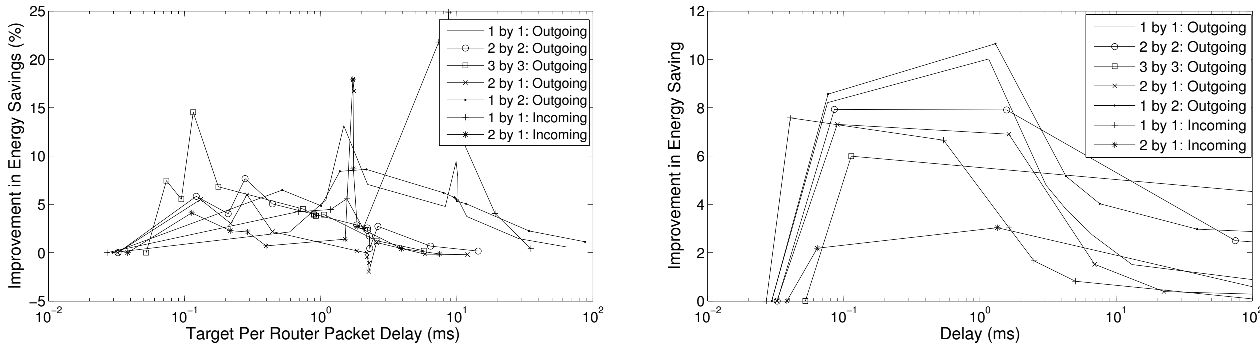
6. ACKNOWLEDGEMENTS

This work was supported by funding from Center for Industrial Information Technology (CENIIT). We thank Martin Arlitt and Anirban Mahanti for helpful discussion on this work, and Rahul Hiran for helping us with the IP-to-AS mappings used in Section 2.1.



(a) Edge (b) Core

Figure 7: Improvements in energy savings (%) on the second router, when using rate switching, under different scenarios and directions (shown in label).



(a) Active/idle toggling (edge) (b) Hybrid (edge)

Figure 8: Improvements in energy savings (%) on the second router, when using the active/idle policy and the hybrid policy, under different scenarios and directions (shown in label).

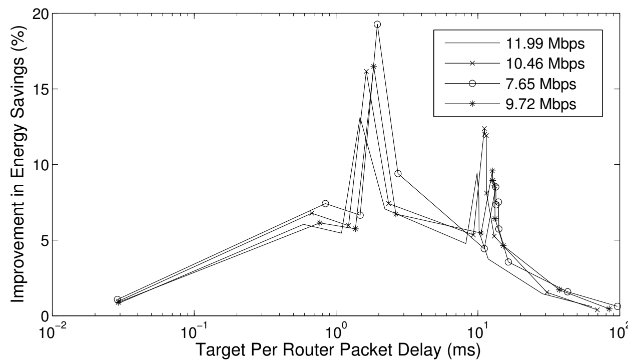


Figure 9: Impact of utilization on the improvements in energy savings on the second router, when using active/idle toggling with different link rates (shown in label).

7. REFERENCES

[1] Ieee 802.3az energy efficient ethernet: Build greener networks. *White Paper from Cisco and Intel* (2011).

[2] Ananthanarayanan, G., and Katz, R. H. Greening the switch. In *Proc. OSDI* (2008).

[3] Barroso, L., and Holze, U. The case for energy-proportional computing. *IEEE Computer 40*, 12 (April 2007), 33–37.

[4] Bolla, R., Bruschi, R., Davoli, F., and Cucchietti, F. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Communications Survey and Tutorials 13*, 2 (2011), 223–244.

[5] Bolla, R., Davoli, F., Christensen, K., Cucchietti, F., and Suresh, S. The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization? *IEEE Communications Magazine 49*, 8 (Aug 2011), 80–86.

[6] Burke, P. J. The output of a queueing system. *Operations Research 4*, 6 (1956), 699–704.

[7] Chiaraviglio, L., Mellia, M., and Neri, F. Energy-aware backbone networks: A case study. In *Proc. IEEE GreenComm* (2009).

[8] Chiaraviglio, L., Mellia, M., and Neri, F. Reducing power consumption in backbone networks. In *Proc. IEEE ICC* (2009).

[9] Cho, K., Mitsuya, K., and Kato, A. Traffic data repository at the WIDE project. In *Proc. USENIX ATC* (2000).

[10] Christensen, K., Reviriego, P., Nordman, B. andBennett, M., Mostowfi, M., and Maestro, J. Ieee 802.3az: The road to energy efficcient ethernet. *IEEE Communications Magazine 48*, 11 (2010), 50–56.

[11] Cleary, J. G. Wand project at university of waikato, nz. In *Proc. HPN: Measurements and Analysis Collaborations Workshop* (1999).

[12] Fan, J., Xu, J., Ammar, M. H., and Moon, S. B. Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-based scheme. *Computer Networks* (Oct. 2004), 253–272.

[13] Gunaratne, C., Christensen, K., and Nordman, B. Managing energy consumption costs in dektop PCs and LAN switches with proxying, split tcp connections, and scaling of link speed. *International Journal of Network Management 15* (Sept. 2005), 297–310.

[14] Gunaratne, C., Christensen, K., Nordman, B., and Suen, S. Reducing the energy consumption of ethernet with an adaptive link rate (alr). *IEEE Trans. on Computers 57*, 4 (April 2008), 448–461.

[15] Gupta, M., Grover, S., and Singh, S. A feasibility study for power management in LAN switches.

[16] Gupta, M., and Singh, S. Greening of the internet. In *Proc. ACM SIGCOMM* (2003).

[17] Gupta, M., and Singh, S. Dynamic ethernet link shutdown for energy conservation on ethernet links. In *Proc. IEEE ICC* (2007).

[18] Hähnel, M., Döbel, B., Völp, M., and Härtig, H. ebond: energy saving in heterogeneous R.A.I.N. In *Proc. e-Energy* (May 2013).

[19] Hanay, Y. S., Li, W., Tessier, R., and Wolf, T. Saving energy and improving TCP throughput with rate adaptation in ethernet. In *Proc. IEEE ICC* (2012).

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

[21] Hohn, N., Papagiannaki, K., and Veitch, D. Capturing router congestion and delay. *IEEE/ACM Trans. on Networking 17*, 3 (June 2009), 789–802.

[22] Huang, T.-Y., Handigol, N., Heller, B., McKeown, N., and Johari, R. Confused, timid, and unstable: Picking a video streaming rate is hard. In *Proc. IMC* (2012).

[23] Kleinrock, L. *Communications Nets: Stochastic Message Flow and Delay*. McGraw Hill, 1964.

[24] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., and Jahanian, F. Internet inter-domain traffic. In *Proc. ACM SIGCOMM* (2010).

[25] Lange, C. Energy-related Aspects in Backbone Networks. In *Proc. ECOC* (2009).

[26] Mao, Z. M., Rexford, J., Wang, J., and Katz, R. H. Towards an accurate as-level traceroute tool. In *Proc. ACM SIGCOMM* (2003).

[27] Meisner, D., Gold, B. T., and Wenisch, T. F. Powernap: Eliminating server idle power. In *Proc. ASPLOS* (2009).

[28] Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., and Wetherall, D. Reducing network energy consumption via sleeping and rate-adaptation. In *Proc. NSDI* (2008).

[29] Pinedo, M., and Wolff, R. W. A comparison between tandem queues with dependent and independent service time. *Operations Research 30*, 3 (1982), 464–479.

[30] Restrepo, J., Gruber, C., and Machoca, C. Energy profile aware routing. In *Proc. IEEE GreenComm* (2009).

[31] Reviriego, P., Christensen, K., Rabanillo, J., and Maestro, J. An initial evaluation of energy efficient ethernet. *IEEE Communications Letters 15* (May 2011), 578–580.

[32] Sandmann, W. Delays in a series of queues with correlated service times. *Journal of Networks and Computer Applications 35* (2012), 1415–1423.

[33] Tucker et al., R. Energy Consumption in IP Networks. *ECOC* (2008).

[34] Weiser, M., Welch, B., Demers, A., and Shenker, S. Scheduling for reduced cpu energy. In *Proc. USENIX OSDI* (1994).

[35] Wierman, A., Andrew, L. L. H., and Tang, A. Power-aware speed scaling in processor sharing systems. In *Proc. IEEE INFOCOM* (2009).

[36] Zhang, B., Sabhanatarajan, K., Gordon-Ross, A., and George, A. Real-time performance analysis of adaptive link rate. In *Proc. IEEE LCN* (2008).

[37] Zhang, G. Q., Yang, Q. F., and Cheng, T. Z. Evolution of the internet and its cores. *New Journal of Physics 10*, 12 (2008), 1–11.

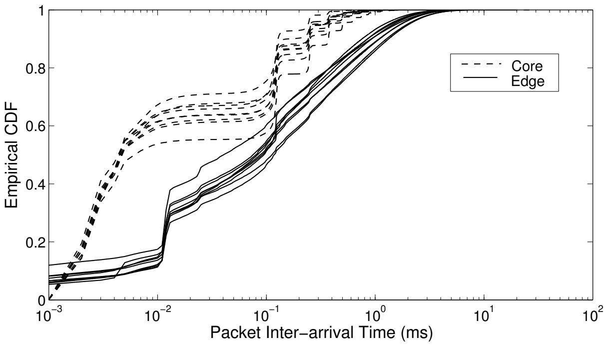
APPENDIX

A. TRACE CHARACTERIZATION

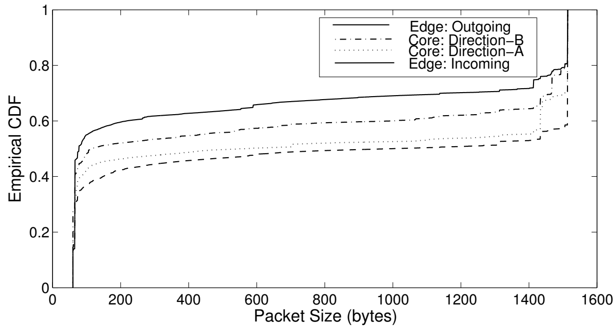
To better understand our results, we must first understand the traffic traces. Figure 10 provides a high-level characterizing of the packet traces. Figure 10(a) shows the empirical Cumulative Distribution Function (CDF) of the packet inter-arrival times seen for ten example days. While common packet sizes and queueing at prior routers result in some frequent inter-arrival times (see curve steps), similar to many other traces collected over shorter time periods, the distributions are exponential in nature, with the general curve shapes being well-fitted by straight lines on lin-log scale.

Figure 10(b) shows the empirical CDF of the packet sizes, with traffic traces broken down by both location and direction. In all cases the observed distributions are bimodal in nature, with most packets being either small (less than 100 bytes) or large (1400-1500 bytes). We call the remaining packets medium sized (100-1400 bytes). While day-to-day variations are observed, the most significant differences are between measurements associated with different location and direction. For example, the outgoing (upstream) traffic at the edge has the largest fraction small packets, and the incoming (downstream) traffic at the edge has the largest fraction big packets. With HTTP being the dominant traffic type [24], this is expected, as the campus users close to the edge likely are consumers. Many of the big packets correspond to data traffic, whereas the small packets going in the opposite direction often will include TCP acknowledgements and HTTP requests.

We next consider the packet-size correlation between back-to-back packets. Table 1 shows the probability of a pair of consecutive packets being of certain packet sizes. In particular, we show the probability that a packet of a certain size category (small, medium, or large) is proceeded by a packet



(a) Inter-arrival times



(b) Packet sizes

Figure 10: Empirical Cumulative Distribution Functions (CDFs) of the packet inter-arrival times and packet sizes, for different traces (shown in labels).

(a) Edge, outgoing

	Small	**Medium**	**Large**
Small	0.39	0.11	0.04
Medium	0.10	0.06	0.03
Large	0.05	0.02	0.20

(b) Edge, incoming

	Small	**Medium**	**Large**
Small	0.24	0.09	0.06
Medium	0.09	0.10	0.07
Large	0.06	0.07	0.24

(c) Core, direction A

	Small	**Medium**	**Large**
Small	0.29	0.07	0.08
Medium	0.06	0.04	0.05
Large	0.10	0.04	0.27

(d) Core, direction B

	Small	**Medium**	**Large**
Small	0.23	0.05	0.07
Medium	0.04	0.02	0.04
Large	0.08	0.03	0.45

Table 1: Packet-size probabilities of back-to-back packets.

belonging to one of the same categories. As expected, we observe significant correlations, with 58-70% of the packets following a packet belonging to the same size category (sum across the diagonals). When we only consider the packets that arrive at the time there is queueing, the bias is even higher. For example, in the case of a single router with transmission rate 1Gbps, approximately 76-77% of the queued packets see a packet of the same size ahead of them in the queue.

When doing this study, we originally wanted to build on traditional two-stage tandem queue models [6, 32]. Unfortunately, these studies typically makes simplifying assumptions that does not match our workloads, including assumptions about exponential service times, independent service times [29], or independent queues [23]. In contrast, the above results show that the real packet traces used in this study include correlations (e.g., Table 1), has bimodular packet size distribution (e.g., Figure 10(b)), and has packet-dependent processing times (proportional to the packet sizes [21], which typically remains fixed along the end-to-end path). For these reasons, we find that some of the observations are different from what would have been predicted by these queuing models.