Saving Energy in LAN Switches: New Methods of Packet Coalescing for Energy Efficient Ethernet

Mehrgan Mostowfi and Ken Christensen
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida
{mostowfi, christen}@cse.usf.edu

Abstract—Small or home office (SOHO) Ethernet LAN switches consume about 8 TWh per year in the U.S. alone. Despite normally low traffic load and numerous periods of idleness, these switches typically stay fully powered-on at all times. With the standardization of Energy Efficient Ethernet (EEE), Ethernet interfaces can be put into a Low Power Idle (LPI) mode during idle periods when there are no packets to transmit. This paper proposes and evaluates a new EEE policy of synchronous coalescing of packets in network hosts and edge routers. This policy provides extended idle periods for all ports of a LAN switch and thus enables energy savings deeper than in the Ethernet PHY only. We evaluate our method using an ns-2 simulation model of a LAN switch. We show that our method can reduce the overall energy use of a LAN switch by about 40%, while introducing limited and controlled effects on typical Internet traffic and TCP.

Keywords—Ethernet switch; synchronized coalescing; Energy Efficient Ethernet; green Internet

I. INTRODUCTION

Networked electronic devices consumed almost 4% (150 TWh) of all the electricity in the U.S. in 2008, about 13% (20 TWh) of which was consumed by network equipment such as network switches and routers [22]. Ethernet, as the dominant wireline technology for LANs, constitutes a significant portion of the energy use of computer networks. The most recent advance in reducing the energy consumption of Ethernet is Energy Efficient Ethernet (EEE) which was standardized as IEEE Std 802.3az-2010 in September 2010 [12]. Products that use EEE are beginning to ship in mid to late 2011.

EEE enables an Ethernet link that is idle to exit Active mode and enter a Low Power Idle (LPI) mode. The power used in LPI mode is significantly less than in Active mode since some components of the physical layer (PHY) can be powered off. The IEEE 802.3az standard does not describe a policy for entering and exiting LPI. A simple policy would be to enter Active mode when packets are queued in the transmit queue in the interface and enter LPI mode when the queue is empty. Such a policy has been found to be inefficient (that is, wasteful of energy) due to excessive transitions between the Active and LPI modes [25]. Each transition incurs a delay and energy use. A policy to batch, or coalesce,

packets was proposed and explored by Christensen et al. in 2010 [6]. Coalescing improves the energy efficiency of Ethernet interfaces in hosts as well as in Ethernet switches. The overall energy consumption of a switch is reduced by EEE as a result of the reduction in energy consumption of individual interfaces of the switch. However, we believe that energy savings by EEE can go beyond the savings in individual interfaces if all the switch ports enter LPI mode at the same time, or synchronously. Synchronized LPI periods provide the opportunity for additional components of the switch to power down since it would be certain that there are no packets in any port buffer, or within the switch fabric, that need to be forwarded. In synchronized coalescing the control of when to coalesce and for how long is moved to the switch (from the host interfaces) and the coalescing periods are synchronized on all the ports of the switch.

The target switches for our synchronized coalescing method are the ones mostly used in households and small offices. This type of switch, which are referred to as SOHO (small or home office) switches hereafter, typically includes 4 to 10 ports and cost less than \$100. Two factors motivated us to propose synchronized coalescing. The first is the typical low utilization of switches in general [9]. The second is that although SOHO Ethernet switches consume only a small amount of energy individually, the number of them deployed in the country is so high that it makes their consumption significant and even a small savings per switch would add-up to a significant overall savings. Using a Kill-A-Watt power meter, the power use of a Linksys EG005W Gigabit Ethernet switch with 4 connected active links was measured as 10 W. We estimate that the current consumption of SOHO Ethernet switches is approximately 7.9 TWh/year based on the number of housing units in the U.S. [26], assuming that about 70% have an Ethernet switch installed, and that each switch is powered on all the time. At the current average electricity cost (\$0.10/kWh) this is a total of about \$790 million per year in electricity use.

While current Ethernet links and switches are mostly 100 Mb/s and 1 Gb/s, we envision that they are likely to evolve to 10 Gb/s in the near future for several reasons including, 1) ever-decreasing prices [20], 2) fast adoption

Figure 1. Transitions between Active and LPI modes in EEE

Figure 2. Close-to-worst case for EEE efficiency

Figure 3. EEE with Packet Coalescing

by vendors [18], and 3) increasing bandwidth requirements of multimedia applications within households (for example audio/video transfer between storage device and player, and LAN-based multi-player video games).

The contributions of this paper are the proposal and explanation of synchronized coalescing and evaluation of its performance trade-offs and effects on typical Internet traffic and TCP through simulation. The remainder of this paper is organized as follows. Section II reviews EEE and previous work that has been done in policies to control EEE. Section III presents a micro-level study of the power use of SOHO Ethernet switches and the opportunity of powering down individual components. Section IV explains the new synchronized coalescing method. Section V is a simulation evaluation of the method. Section VI describes related work. Section VII summarizes, describes future work, and estimates the potential energy savings that could be gained by large-scale deployment of the presented methods.

II. OVERVIEW OF ENERGY EFFICIENT ETHERNET (EEE)

EEE brings the energy consumption of Ethernet links closer to the ideal consumption, which is directly proportional to the utilization of the link. Estimates show that using EEE in all current 1 Gb/s edge links in both residential and commercial buildings and network equipment links within residences could save about \$180 million/year in the U.S. alone [6]. Two modes are defined in EEE; Active mode and Low Power Idle (LPI) mode. In Active mode the link is powered-on to transmit packets. When there are no more packets to transmit, the link can enter LPI mode in which the physical layer is powered off and elements in the receiver are stopped. The arrival of a packet to the link can result in the link to wake up in a few microseconds to resume packet transmission (Figure 1). In this figure, T_s is the time

needed to enter LPI mode and T_w is the time needed to return to Active mode. During the T_w , T_s and T_r periods the link consumes full power, while during T_q only almost 10% of the full power consumption of the link is needed [25]. The refresh cycle of duration T_r is a periodic link activity to maintain the alignment of receiver elements to channel conditions. It can be assumed that the link uses the same power as in Active mode during transitions [25]. The minimum T_w and T_s for 10GBASE-T links are 4.48 and 2.88 μ s respectively [12]. The transition times are relatively high compared to the transmission time of 1.2 μ s for a 1500-byte packet at 10 Gb/s.

A. Improving the efficiency of EEE

The transition times of EEE are large compared to the transmission time of a packet. For instance, if the link wakes up to transmit a single 1500 byte packet, it would spend 1.2 μ s transmitting the packet and 7.38 μ s for transitioning from LPI mode to Active and back. This means that only about 14% of time is dedicated to transmitting the packet and the rest to the transitions. This inefficiency of EEE was first explored by Reviriego et al. [25] in 2009. EEE can be most efficient when packets arrive back to back in bursts. As a result, only one sleep and one wakeup transition is required per burst which makes the percentage of time the interface spends in active mode close to the link utilization. This best case often occurs in the form of file downloads using TCP where large blocks of data are burst onto a link from a server to a client at a high rate.

Conversely, the worst case happens when packets arrive with a fixed inter-arrival time and a spacing greater than the wake and sleep transition times. As a result, one wake and one sleep transition would be required for transmission of each packet resulting in inefficient operation. A close-to-worst case traffic scenario occurs when TCP ACKs are being returned from a client to a server. TCP ACKs are typically small packets and are spaced-out evenly (Figure 2).

The inefficiency of EEE can be reduced by coalescing the outgoing packets into bursts thus decreasing the number of necessary transitions to one per burst. Packet coalescing for EEE is demonstrated and studied in [6]. EEE with packet coalescing is depicted in Figure 3. As shown in this figure, when all the packets in the transmit queue (or buffer) are transmitted and the buffer becomes empty, the link is put to LPI mode after a sleep transition which takes T_s time. The packets which arrive thereafter are not transmitted immediately but are buffered into a coalescing buffer. When a maximum time passes from the arrival of the first packet to the coalescing buffer, or the number of buffered packets reaches a pre-defined maximum, the link exits the LPI mode, which takes T_w time. All the coalesced packets are transmitted in a single burst. Refresh periods are omitted in Figure 3, since $T_q \gg T_r$.

III. SWITCH ENERGY USE AND TRANSITION TIMES

To determine the possible energy savings from synchronization of LPI periods between all ports in a switch, it is necessary to answer the following three questions:

- 1) Which components of the switch can be powered down?
- 2) How much reduction in total switch power use can be achieved by powering down these components?
- 3) What are the required times to transition these components from fully-powered to powered-down mode and back?

The main component of a typical SOHO Ethernet switch is a single CMOS switch chip. D-Link DGS-1008G, Linksys EG008W, Netgear GS608, and Trendnet TEG S8 are common examples of such switches (all are Gigabit Ethernet switches). The first uses Vitesse VSC7388 SparX-G5 [27], and the rest use Broadcom BCM5398 chips [5], both of which are switch-on-a-chip ICs that include the switching fabric, Ethernet port blocks, interfaces to external CPUs, memories, and the layer 2 packet header processor. The chip is connected to the copper interfaces and optionally to external memories and CPUs. The switching fabric is a high performance bus, shared among all the port blocks and the processor. Each port block consists of a copper PHY, a MAC and ingress and egress packet queues. The packet forwarding tables are maintained in the processor's memory and registers. Packets enter through PHY interfaces, are passed to the MAC, are put in the port block's egress queue, and are then put on the bus. The header of the packet is analyzed by the packet processor and either the forwarding port is determined, or it is filtered. The forwarding tables are modified accordingly at this point, if needed. The packet is then put in the ingress queue of the forwarding port. Then it goes through the MAC and the PHY to the outgoing interface.

We believe that a synchronized idle period could allow the switch chip to sleep while maintaining internal state. It is important, however, to empty all the ingress and egress queues prior to power down in order not to lose any packets. This answers question 1.

To answer question 2, the power use of a 10 Gb/s SOHO Ethernet switch must first be determined. Since such switches are yet to be manufactured and marketed for SOHO use, we estimate their power use as follows. We measured the average power use of a Linksys EG005W Gigabit switch with 4 active links as 10 W. We envision 10 Gb/s switches becoming commonplace around 2016, roughly 10 years after the standardization of 10GBASE-T, which is the same time span from the standardization of Gigabit Ethernet in 1999 to 1 Gb/s becoming status quo in 2009. Using 100 W consumption as the base (linear relation with capacity increase) and 20% yearly improvement in router power efficiency [2], the power use of future status quo switches over 10 years would

Figure 4. System view of synchronized coalescing [3]

be roughly 10 W (that is, the same as today). The CPU and memories of the linecards of a high-end enterprise router consume more than 50% of the linecards power [7]. We assumed that the breakdown of the power use of a 10 Gb/s SOHO switch will roughly correspond to the breakdown of a single line card of an enterprise router with the CPU and memories embedded in the switch chip. Therefore we assumed that by powering down the chip the overall power use decreases to 50% overall.

To answer question 3, we use the transition times between C0 and C6 states of an Intel i5 multi-core processor (2 ms) as a conservative upper bound of the transition time for both the chip and possible external CPUs [4]. The synchronized LPI periods should be at least twice this time to allow the chip to transition to the low-power state and back to the full-power state.

IV. THE SYNCHRONIZED COALESCING METHOD

In synchronized coalescing, all the links connected to a LAN switch are stopped from sending any traffic at the same time for a fixed period of time. The network interfaces at both ends then enter LPI mode automatically, and the previously described components of the LAN switch can be turned off or put into a sleep mode. Synchronized coalescing is specifically intended for SOHO Ethernet switches since these are the most lightly utilized switches in the network with many idle periods. Therefore, there is a good opportunity of saving energy in these switches while minimizing any possible adverse effect on the performance of the network. Synchronized coalescing builds on the Pause Power Cycle (PPC) idea that was proposed and prototyped prior to the standardization of EEE in 2008 [3].

Figure 4 shows a high-level system view of how synchronized coalescing works. This is similar to the configuration used in [3]. Synchronized coalescing is implemented in the LAN switch and works as follows: a notification message (referred to as Pause Notification hereafter) indicating that

Figure 5. FSM for simple synchronized coalescing

the interface connected to the port must not send any traffic for an arbitrary interval is sent by the switch on all the ports. The switch then enters a low-power mode in which the components mentioned earlier are powered off or put into a sleep state. When the interval elapses, the switch powers up to a fully operational state and resumes servicing packets (ON state). The time that the switch spends in OFF and ON states are called T_{off} and T_{on} respectively. Based on T_{off} and T_{on} , a parameter called Duty Cycle (D) is defined as $T_{on}/(T_{on} + T_{off})$. By fixing T_{off} and D , T_{on} can be determined as $(D \cdot T_{off})/(1 - D)$.

One of the mechanisms that can be used to notify the NIC to stop sending any traffic for a period of time is the flow control mechanism known as PAUSE frames defined in Ethernet standard. However, the traffic stopping part of synchronized coalescing could be implemented using other notification mechanisms as well. The implementation of synchronized coalescing in real switches requires some sort of Pause Notification to be supported by MAC or PHY layers. PAUSE frames notify the NIC to temporarily stop the flow of traffic (except for MAC control frames) for a certain period of time [14]. PAUSE frames are intended to allow an end of a connection to recover from a congestion state by temporarily stopping the other end from transmitting more packets. By setting MAC Control Parameters field to T_{off} value, PAUSE frames can be used to make sure that no traffic will be received and powering off will not cause packet loss.

A. Simple synchronized coalescing

The simplest version of synchronized coalescing is when the switch stays in ON state for a fixed period of time, enters the OFF state, stays there for a fixed period of time and the process repeats. Simple synchronized coalescing is described by a Finite State Machine (FSM) in Figure 5. The switch is fully operational in the ON state, while the links connected to the switch are paused and the switch is powered down (or sleeping) in the OFF state. Two timers are defined; TON and TOFF which maintain the time spent in ON and OFF states. The initial values of the timers are T_{on} and T_{off} , respectively.

B. Adaptive coalescing

The use of simple synchronized coalescing results in large increases in per packet delay especially when the aggregate load on the link is high or when sudden bursts of packets

Figure 6. FSM for adaptive coalescing

flow to the switch. This effect will be demonstrated and explained further in Section V. To reduce this effect, a modification to simple synchronized coalescing is made which regulates the transition to the OFF state based on the number of packets received while in the previous ON state. Simple synchronized coalescing with this modification is called adaptive coalescing hereafter. Adaptive coalescing is described in the FSM in Figure 6. This FSM has the same states as the FSM in Figure 5. A new array and a new variable are defined as following:

- *pktCount*: An array of the size equal to the number of links connected to the switch. Each element of *pktCount* stores the number of packets received from the corresponding link. Index i is for the *pktCount* and ranges from 1 to the number of links.
- *thresh*: The threshold which is compared to the maximum of all *pktCount_i*s to determine if transition to OFF state should be made.

Upon start or reset, the switch enters ON state, TON is set to its initial value T_{on} and starts to count down. Also, all elements of *pktCount* are initialized to 0. When TON expires, two cases can happen: 1) the maximum over all elements of *pktCount* is greater than or equal to *thresh* (this to handle uniformly distributed load among all ports, as well as heavy traffic or sudden burst on a single port), in which Pause Notifications are sent on all the links, TOFF is reset, starts to count down and the switch enters OFF state, or 2) otherwise; in which case TON is reset to its initial value and starts to count down, all elements of *pktCount* are set to 0 and the switch remains in ON state for another ON period. Upon expiration of TOFF, TON is reset to its initial value and starts to count down, all elements of *pktCount* are set to 0, the switch returns to ON state and the entire procedure repeats.

V. EVALUATION BY SIMULATION

We evaluated the energy savings and performance trade-offs of our methods using an ns-2 [21] simulation model. To model synchronized coalescing in ns-2 we added two timers to the wired PHY module of LAN networks to stop and resume passing packets to the lower layer for fixed periods of time. We synchronized ON and OFF periods among all

the links in a LAN by using static variables shared among all the PHYs.

The experimental factors of the system configuration were bandwidth and the RTT of the server-edge router, switch-edge router link, and switch-client links. The factor of the traffic flows was data flow type - arbitrary packet distribution or FTP. The factors of synchronized and adaptive coalescing were T_{off} , D , and $thresh$. The response variables were the switch ON time, download time (for file transfer), and average per packet delay (for arbitrary packet distribution flows). From the switch ON time, energy savings could be calculated.

A. Design of experiments

We modeled the network configuration shown in Figure 4 in ns-2 for all of our simulation experiments. The server and the edge router were connected by a simulated link with 100 μ s RTT. Wherever the transport protocol was TCP, a maximum congestion window size of 60 packets was used which corresponds to the default maximum TCP window size in most Microsoft Windows distributions. The switch and the edge router were in the same LAN with 2 μ s RTT. The clients and the switch are in another LAN with the same 2 μ s RTT. This roughly corresponds to a LAN whose clients, switch and edge router are located in the same building.

We seek to answer the following questions with an experimental evaluation of a simulation model of synchronized coalescing:

- 1) How does synchronized coalescing handle high traffic loads and sudden packet bursts?
- 2) How do the parameters of adaptive coalescing affect switch ON time?
- 3) What are the performance tradeoffs of adaptive coalescing?
- 4) How does adaptive coalescing affect real applications such as file download?

Three experiments were designed; they were 1) high load effect experiment, 2) threshold experiment, and 3) file download experiment. Each experiment was designed to answer the questions listed above.

High load effect experiment. This experiment was designed to study the effects of high loads and sudden bursts of packets on the performance of synchronized coalescing (Question 1). Packets flow from Client 1 to the switch over UDP. Packet interarrival time was exponentially distributed. At large time scales network traffic is likely self-similar [16], however a Poisson distribution remains a reasonable first-order approximation in cases which the traffic is highly aggregated and the traffic sample is taken in sub-second (small) time spans [15]. The size of the packets was fixed to 1500 bytes (maximum Ethernet MTU, compatible with bridged/switched 1 Gb/s systems). Simple synchronized coalescing was enabled on the switch with two duty cycles of 10% and 50%. T_{off} was fixed to 100 ms. Since 100 ms

is approximately the human response time, this is likely the upper bound of tolerable delay. A T_{off} of 100 ms limits the relative time and consequent energy waste of transitions to less than 2% of the sleep time. The offered load was varied by manipulating the mean interarrival time of the packets. The response variables of interest were the switch ON time and average per packet delay.

Threshold experiment. This experiment was designed to study the effects of the threshold used in adaptive coalescing on packet delay and switch power use (Questions 2 and 3). A packet flow (with exponential interarrival times) over UDP from Server to Client 1 with a fixed packet size of 1500 bytes and variable offered load was modeled as in the previous experiment. T_{off} was again fixed to 100 ms. Adaptive coalescing was enabled on the switch with 10% duty cycle. Two thresholds of 1000 and 5000 packets were used. The former (low threshold) corresponds to almost 10% of the link capacity (equal to the duty cycle). The latter (high threshold) is almost half of the link capacity. The response variables of interest were the switch ON time and average per packet delay.

File download experiment. This experiment was designed to study the effects of T_{off} and the adaptive coalescing threshold on file download over TCP (Question 4). A 125 Megabyte file (corresponding to the file size of a small video clip) was downloaded by Client 1 from the Server using FTP. Adaptive coalescing was enabled on the switch with a 10% duty cycle. Three adaptive coalescing thresholds of 100, 500 and 1000 packets were used to show the effects of various thresholds on the response variables. T_{off} was varied to cover the range from 0 to slightly more than the 100 ms. The response variables of interest were the file download time and the switch ON time.

B. Experiment results and discussion

The experiment results were as follows. For the *high load effect experiment* the average per packet delay as a function of offered load on the link is shown in Figure 7. It can be observed that when the load is more than the duty cycle, the average delay starts to rapidly increase due to instability of the buffer queue. Note that packets are delayed in the NICs connected to the switch, not inside the switch. This is due to halting of transmission from the NICs temporarily during T_{off} periods while the packets are still being generated. The same phenomenon happens if there is a sudden burst in the traffic, even though the overall load is less than the duty cycle. The threshold for skipping OFF periods in adaptive coalescing is an effective way to overcome this shortcoming. The switch ON time is the same as the duty cycle regardless of load.

The results for the *threshold experiment* in Figure 8 show the percentage of time the switch spends in ON state as a function of load. Using the high threshold, the ON time of the switch is less than 10% different than the load-

Figure 7. High load effect experiment results – packet delay

Figure 8. Threshold experiment results - ON time

Figure 9. Threshold experiment results - packet delay

proportional which is the ideal ON time for the switch. However, the ON time is more when the low threshold is used. Using the low threshold, when the load exceeds the duty cycle (10%) the ON time ascends quickly to 100% and stays at this level as the load increases. This sudden increase is because the number of arriving packets during ON periods exceeds the threshold. The “steps” seen in both traces are

also because the number of arriving packets exceeds the threshold. At the points that the ON time increases to the next step, the number of packets arriving during some of the ON periods exceeds the threshold. After each increase to a given step, the ON time stays roughly the same until the next increase to a new step. This is because while the number of arriving packets increases, the increase is not so high to cause any more ON periods to have more arriving packets than the threshold. Therefore, the ON time does not change in these intermediate loads. Although adaptive coalescing brings the ON time close to load-proportional, it introduces a delay to some packets. The reason is that the switch is sometimes OFF and unable to service packets while packets keep arriving to the attached devices to the switch. These packets are queued and delayed. Figure 9 shows the average packet delay caused by adaptive coalescing as a function of offered load for the two thresholds. Steps are also seen in this figure as the load increases. Each step corresponds to an ON time step in Figure 8. The reason is the same as what described for ON time. The average per packet delay is less than half of the human response time (50 ms) which has minor negative effect for non-delay-sensitive applications [23]. The sudden drop in the delay at 10% load by using the low threshold happens when the number of packets arriving to ON periods exceeds the threshold constantly, which causes the switch to stay ON and service the packets immediately.

If no coalescing were used, it would take about 1 s to download an entire 125-Megabyte file over the 1 Gb/s bandwidth (of the link between the edge router and the server through the Internet). Figure 10 shows the download time of the file for the *file download experiment* when using simple synchronized coalescing and adaptive coalescing with 10% duty cycle and three thresholds. Using simple synchronized coalescing with 10% duty cycle is as if the capacity of the switch is cut to 10%, which makes the capacity of the switch links equal to the bandwidth of the server through the Internet (1 Gb/s). This would suggest no increase in file download time. However, it is seen in Figure 10 that this is only the case when T_{off} is very small (less than 3 μ s). As T_{off} increases, the download time increases exponentially until it becomes stable at about 10 s which is about one tenth of the capacity of the link between the edge router and the server. Instrumentation of our simulation showed that this counterintuitive effect is due to side effect with the TCP congestion control mechanism. If T_{off} is longer than the time needed to transmit a full congestion window, the transmission pauses from the server until the window frees up in the next ON period.

This effect can be controlled by reducing T_{off} . As explained in Section III, however, T_{off} cannot be set to less than 4 ms due to transition times. Too small T_{off} times, although more than the minimum, drastically increase the power waste due to relatively high transition times.

Figure 10. File download experiment results - download time

Figure 11. File download experiment results - ON time

Adjusting the threshold instead, can be used to control this effect temporarily while not sacrificing the beneficial length of T_{off} . As seen in Figure 10, if the threshold is set to roughly the same number of packets that could be transmitted at full rate in a T_{on} , (for instance 100 packets for the duty cycle of 10%, T_{off} of 10 ms and T_{on} of 1.11 ms) the threshold is exceeded in case of a file download and makes the transfer time the same as it would be if no coalescing method were used. As a consequence, the switch would stay ON for the entire download time as depicted in Figure 11. Figure 11 shows the percentage of time the switch spends in ON state as a function of T_{off} for the duration of the complete download of the file over FTP. The same three thresholds as in Figure 10 are shown. As expected, the switch stays ON for 10% of the time when the threshold is larger than what transmitted in an OFF period. When the threshold is exceeded, the switch stays ON constantly.

From the results of our experiments, it seems that adaptive coalescing can save a significant amount of energy on a 10 Gb/s SOHO Ethernet switch with only small trade-off in reduced performance provided that the following conditions are met:

- The Duty cycle (D) is set to a value which makes the switch spend most of the time in OFF state.
- The control variable T_{off} is set to less than the human response time and reasonably more than the switch transition times.
- The threshold ($thresh$) is set to a value which detects file transfers and bursts well.

We proposed 10%, 100 ms, and 1000 packets for D , T_{off} , and $thresh$ respectively, and showed that these values provide about 80% sleep time for the switch while introducing a small reduction in performance (less than 50 ms of average per-packet delay and almost no reduction in the case of a sudden traffic burst with the expense of less energy savings).

VI. RELATED WORK

The first standard which included a coalescing method for the purpose of saving energy was in wireless networks in late 1990s. In 1999, the Power Saving Mechanism (PSM) for Distributed Coordination Function (DCF) in IEEE 802.11 wireless networks was standardized [13]. In PSM, time is divided into equal beacon intervals. Packets destined for a wireless station are coalesced in the preceding station until they are transmitted during the first beacon interval. Proposed improvements to PSM modified the coalescing criteria to improve its performance under specific conditions or for a special type of traffic ([24] for instance).

The most recent coalescing method for wireless mobile devices is Catnap [8]. Catnap introduces a proxy on the wireless router connecting the mobile device and the Internet. The proxy gathers incoming packets and bursts them to the wireless device on scheduled intervals letting the device sleep between the bursts.

Research in reducing energy use of wired networks started in the early 2000s [11]. The first proposed use of packet coalescing to reduce energy use of Ethernet switches was Dynamic Ethernet Link Shutdown (DELS) [10]. DELS makes a decision to sleep a LAN port based on the statistics of the packets already buffered in the port transmission queue (for the given link). Packets arriving while a port is in sleep mode are buffered to be transmitted when the sleep period is over. Another method called Buffer-and-Burst (B&B) buffers packets at network edge routers in order to enable some of the routers inside the network to sleep [19]. With B&B, each ingress router of the network buffers packets for a certain time interval, and then bursts the entire buffer into the network. This enables all the egress routers to sleep between bursts.

The most recent coalescing method for wired links is (unsynchronized) packet coalescing for EEE [6]. Packet coalescing for EEE is comparable to receive side coalescing [17], which is already in use in many Ethernet NICs. Receive side coalescing reduces the interrupt overhead on the main CPU by issuing one interrupt per many received packets instead of one for each packet. Our work builds on previous

work in packet coalescing for EEE by extending the energy saving opportunity from the Ethernet link to both the link and within the switch.

VII. SUMMARY AND FUTURE WORK

We have shown how periodically powering down the components of a SOHO Ethernet switch after pausing the traffic from all the connected links in a synchronized manner can reduce the power use of the switch to almost proportional to the offered load. We have also shown that by using adaptive coalescing with a suitable threshold, power use of close to load-proportional is achievable with an average delay of less than half that of the human response time (of 100 ms). Moreover, we have shown that adaptive coalescing introduces virtually no added delay to file transfer over TCP by keeping the switch ON for the entire file download time. Based on our results, we believe that adaptive coalescing is a viable method for significantly reducing the energy consumption of Ethernet switches without an excessive performance penalty. Since the aggregate load on a SOHO Ethernet switch is typically very low, we assumed the average aggregate utilization of all links to be 7 to 10% on average. Based on the results shown in Figure 8, about 80% OFF time can be achieved in a LAN switch using adaptive coalescing at these loads. Using the power consumptions in ON and OFF states, and the total number of SOHO switches estimated in Section I, we estimate the potential energy savings that can be obtained by deploying adaptive coalescing on all the future SOHO switches to be approximately 3.5 TWh/year in the U.S. alone. As future work we need to consider:

- Measuring the effects of synchronized coalescing on interactive applications (such as gaming) and web surfing.
- Analytical and simulation evaluation of synchronized and adaptive coalescing when it is extended to multiple hops at the same time.

ACKNOWLEDGMENT

This material is based on work partially supported by Cisco Collaborative Research Initiative (Mostowfi). We also thank the anonymous reviewers for their insightful comments that have resulted in significant improvements to this paper.

REFERENCES

- [1] J. Aweya, D. Montuno, and M. Ouellette, "Limitations of current Ethernet switch architectures for enhanced flow control and service differentiation," *Intl. J. of Network Management*, vol. 19, pp. 153-174, May 2009.
- [2] J. Baliga, K. Hinton, and R. Tucker, "Energy consumption of the Internet," *Proc. COIN/ACOFI*, Jun. 2007.
- [3] F. Blanquicet, "PAUSE Power Cycle: A new backwards compatible method to reduce energy use of Ethernet

- switches," White paper, Ethernet Alliance, Apr. 2008.
- [4] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli, "Theoretical and technological limitations of power scaling in network devices," *Proc. ATNAC*, pp. 37-42, Oct./Nov. 2010.
- [5] Broadcom Corporation, "8-GE port switch with integrated SerDes," BCM5398 Datasheet, 2006.
- [6] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro, "IEEE 802.3az: The road to Energy Efficient Ethernet," *IEEE Communications*, vol. 48, no. 11, pp. 50-56, Nov. 2010.
- [7] G. Epps and A. Baldini, Talk in Cisco Green Research Symposium, Mar. 2008.
- [8] R. Fahad, P. Dogar, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," *Proc. 8th MobiSys*, 2010.
- [9] M. Gupta, S. Grover, and S. Singh, "A feasibility study for power management in LAN switches," *Proc. 12th IEEE ICNP*, pp. 361-371, Oct. 2004.
- [10] M. Gupta and S. Singh, "Dynamic Ethernet Link Shutdown for power conservation on Ethernet links," *Proc. IEEE Intl. Conf. on Communications*, Jun. 2007.
- [11] M. Gupta and S. Singh, "Greening of the Internet," *Proc. ACM SIGCOMM*, pp. 19-26, Aug. 2003.
- [12] IEEE P802.3az, Energy Efficient Ethernet Task Force, 2010.
- [13] IEEE Std 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 1999.
- [14] IEEE Std 802.3x, "Standards for Local and Metropolitan Area Networks," 1997.
- [15] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary Poisson view of Internet traffic," *Proc. IEEE INFOCOM*, Nov. 2004.
- [16] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM TON*, vol. 2, pp. 1-15, Feb. 1994.
- [17] S. Makineni et al., "Receive side coalescing for accelerating TCP/IP processing," *Proc. 13th HiPC*, Dec. 2006.
- [18] R. Myslewski, "Intel dubs 2009 'The year of 10Gb Ethernet'," *The Register*, Mar. 2009.
- [19] S. Nedeveschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via rate-adaptation and sleeping," *Proc. NSDI*, Apr. 2008.
- [20] H. Newman, "Falling 10GbE prices spell doom for fibre channel," *EnterpriseStorageForum.com*, Jun. 2009.
- [21] The Network Simulator - ns-2, 2008. URL: <http://www.isi.edu/nsnam/ns/>
- [22] B. Nordman, Talk in EEDN PAC meeting, California Energy Commission, Sep. 2009.
- [23] V. Paxson, "Some perspectives on the performance impact of link-speed switching outages," Presentation at IEEE 802.3 meeting, Jul. 2007.
- [24] C. Poellabauer and K. Schwan, "Energy-aware traffic shaping for wireless real-time applications," *Proc. 10th IEEE RTAS*, pp. 48-55, May 2004.
- [25] P. Reviriego, J. Hernandez, D. Larrabeiti, and J. Maestro, "Performance evaluation of Energy Efficient Ethernet," *IEEE Communication Letters*, vol. 13, no. 9, Sep. 2009.
- [26] USA QuickFacts from the U.S. Census Bureau, URL: <http://quickfacts.census.gov/qfd/states/00000.html>
- [27] Vitesse Semiconductor Corporation, "SparX-G8™-8-Port gigabit Ethernet integrated PHY switch," VSC7388 Datasheet, Mar. 2006.