Greening the Switch

Ganesh Ananthanarayanan
University of California, Berkeley
ganesha@cs.berkeley.edu

Randy H. Katz
University of California, Berkeley
randy@cs.berkeley.edu

Abstract

Active research is being conducted in reducing power consumption of all the components of the Internet. To that end, we propose schemes for power reduction in network switches – Time Window Prediction, Power Save Mode and Lightweight Alternative. These schemes are adaptive to changing traffic patterns and automatically tune their parameters to guarantee a bounded and specified increase in latency. We propose a novel architecture for buffering ingress packets using *shadow* ports.

We test our schemes on packet traces obtained from an enterprise network, and evaluate them using realistic power models for the switches. Our simple power reduction schemes produce power savings of upto 32% with minimal increase in latency or packet-loss. With appropriate hardware support in the form of Wake-on-Packet, shadow ports and fast transitioning of the ports between its high and low power states, these savings reach 90% of the optimal algorithm's savings.

1 Introduction

The energy efficiency of Internet equipment is important for economic as well as environmental reasons. Networking equipment is experiencing an increase in performance – switches with speeds of 10 Gbps are in the market – that has caused a substantial increase in its power consumption. Studies estimate the USA's network infrastructure uses 24 TWh per year [7], or $24 billion. This includes network switches, access points, end-node NICs and servers. Efforts are underway to reduce the power consumption of all the components of the Internet leading to standards like EnergyStar and an IEEE task force for energy efficient ethernet [4]. As part of the larger goal, in this paper, we propose power reduction schemes for network switches.

Networks are generally provisioned for peak loads due to performance reasons. But network traffic has been observed to have two characteristics - (i) Bursty with long interspersed idle periods [9], and (ii) Diurnal variations in loads, e.g., web servers [16]. This results in heavily under-utilized equipment during non-peak times. Studies have shown that network utilization is under 30% even for backbone networks [5].

The theme of our power reduction schemes is to trade some performance (latency and packet-loss) for significantly reduced power consumption. We propose three schemes – Time Window Prediction, Power Save Mode and Lightweight Alternative. Our schemes assume some hardware characteristics that enhance power savings. Some of them are already available today and we aim to demonstrate the utility of the rest, and make a case for their incorporation in future switch designs.

Overall, we make the following contributions: Firstly, we present two simple power reduction schemes – Time Window Prediction (TWP) and Power Save Mode (PSM) – that we believe are easy to implement on switches. The schemes operate stand-alone on a switch and hence can be incrementally deployed. Secondly, we analyse the trade-off between performance and power consumption. In doing so, we introduce and demonstrate the value of pre-specified and bounded performance degradation. Finally, we make a set of recommendations for switch designs viz., lightweight alternative, shadow ports, Wake-on-Packet and low-powered modes in switch ports with fast transitioning.

The rest of the paper is organized as follows. Section 2 describes the switch's architecture. Section 3 describes the Time Window Prediction and Power Save Mode schemes, and Section 4 evaluates them. We describe the Lightweight Alternative in Section 5. Related work is presented in Section 6. We conclude in Section 7.

2 Switch Architecture

In this section, we present the architecture of the switch including the power model and the buffering capabilities.

Parameter	Value
$Power_{fixed}$	60W
$Power_{fabric}$	315W
$Power_{line-card}$ (first card)	315W
$Power_{line-card}$ (subsequent cards)	49W
$Power_{port}$	3W
$Power_{port}$ (idle)	0.1W
Port-transition – Power	2W
Port-transition – Time (δ)	1ms to 10ms

Table 1: **Parameters used in the Power Model**

2.1 Power Model

A typical modular switch's power consumption is divided among four main components – chassis, switching fabric, line-cards and ports. The chassis includes the cooling equipment, e.g., fan, among other things and its power consumption is denoted as $Power_{fixed}$. The switching fabric is responsible for learning and maintaining the switching tables and its power consumption is denoted as $Power_{fabric}$. The line-card maintains buffers for storing packets. Ports contain the networking circuitry. The line-card acts as a backplane for multiple ports and forwards packets between the switching fabric and ports. Modern line-cards can support 24, 48 or 96 ports. Note that a switch can contain multiple line-cards. We denote the line-card's power consumption to be $Power_{line-card}$ and every port's power consumption to be $Power_{port}$. Hence the total power consumption of the switch can be viewed as, $Power_{switch} = Power_{fixed} + Power_{fabric} + numLine * Power_{line-card} + numPort * Power_{port}$.

We use values for $Power_{fixed}$, $Power_{fabric}$ and $Power_{line-card}$ from the Cisco Power Calculator [2] corresponding to the Catalyst 6500 switch (we discuss $Power_{port}$ shortly). We assume that the power consumed by this switch is indicative and typical of similar products (Table 1). TWP and PSM schemes concentrate on intelligently putting ports to sleep during idle periods; other components of the switch are assumed to be powered on always. Hence, for a switch with four line-cards and 192 ports, the maximum saving is 39.4%.

2.2 Port Design

We define a two-state Markov model for a port's power states, transitioning between a high-power and a low-power state. The transition is assumed to take a finite time δ. Every port consumes 3W in its high-power state [3]. Consistent with prior work [15], for values of transition time δ, and power-consumption in low-power state, we use values from the wireless domain (Table 1).

Buffering: Each port is bi-directional and has packets flowing into the switch (ingress) and away from the switch (egress). Egress packets are buffered automatically when the port is in low-power state. When a port transitions back to high-powered state, it processes the buffered packets. Ingress packets, on the other hand, have to be received by the port and forwarded to the buffers for further processing. Ingress packets are lost in current switches when they arrive at a port in its low-powered state. To address this, we propose the *shadow* port. A shadow port receives ingress packets if any of the conventional ports are in low-power state. A shadow port's hardware is similar to normal ports.

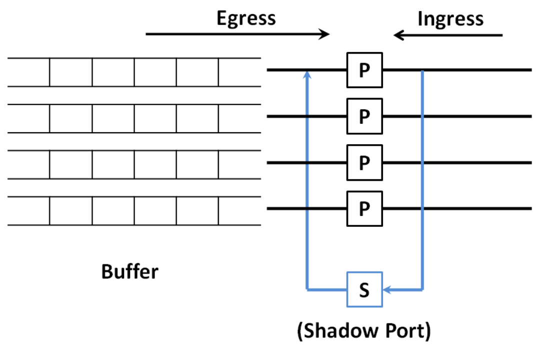

Figure 1: **Shadow Port for a cluster of size 4**

Each shadow port is associated with a cluster of normal ports. Since its power consumption is the same as that of a normal port, savings can be achieved if atleast two of the normal ports in a cluster are in low-power state in the same time. Figure 1 shows a conceptual diagram of the shadow port's architecture for a cluster of size 4. Shadow ports receive only one incoming packet at a time. If packets arrive simultaneously at multiple conventional ports in their low-powered state, all but one of them are lost. While ingress packets are lost due to simultaneous arrivals on the shadow port, egress packets are lost due to buffer overflow.

Wake-on-Packet: During sustained idle periods, to avoid the overhead of unnecessary transitions to the high-powered state, we assume a Wake-on-Packet (WoP) facility. Using this facility, a port automatically transitions from the low-powered state to the high-powered state on arrival of a packet. This packet is lost if it is an ingress packet; egress packets are all buffered. Shadow ports do not receive ingress packets for a port that has put itself to indefinite sleep relying on WoP.

While timer-driven transitions are still valuable as they enforce a minimum sleeping duration even in the presence of traffic flow, WoP helps the ports take better advantage of sustained idle periods. The hardware support for WoP would be similar to Wake-on-LAN [6].

3 Time Window Prediction

The Time Window Prediction (TWP) scheme observes the number of packets crossing a port in a sliding time-window of t_o units and assumes that to be an indication of traffic in the next window. If the number of packets in this time-window is below a threshold τ_p, the switch powers down the port for t_s units (sleep time window). Packets that arrive at the port when it is powered down are buffered (refer Section 2.2).

Adaptive Sleep Window: A good prediction function would reduce erroneous sleeps and the consequent increase in latencies. Neverthless, we believe that the performance of the scheme should not entirely rely on the accuracy of the prediction function. Egress packets that arrive at a port when it is asleep are buffered and sent after the port wakes up. This causes an increase in latency. Note that this in addition to the latency incurred due to various factors along its path. For the sake of brevity, we interchangeably refer to this increase in latency as simply latency. Ingress packets are handled by the shadow port and incur no extra latency.

TWP is supplied with a per-port bound on the tolerable increase in per packet latency, L, and it dynamically adapts its sleep-window t_s to meet the latency bounds. Note that t_s is also automatically increased in times of low network activity and this increases the power savings. Table 2 describes the adaptive Time Window Prediction scheme. Packet latency can be calculated using its time of arrival and time of processing. *avg-lat* is the running average for the per-packet increase in latency over a long term. The weights w_1 and w_2 were each set to 0.5 for the evaluations. This ensures that the scheme is adequately sensitive to changes in traffic patterns. The lower-bound for sleeping τ_s is set to twice the transition time δ to ensure that the overhead of switching states is not higher than the power saving because of the sleep. TWP does not examine multiple observation time windows. The size of the sleep window, and hence ratio of the observation time window to the sleep window, is adaptively adjusted. This is equivalent to observing across multiple time windows.

We also measured the results of our algorithm if the ports supported a Wake-on-Packet (WoP) facility. If there are no packets in multiple consecutive t_o windows, the port goes into indefinite sleep and wakes up on an incoming packet.

Power Save Mode (PSM): PSM is a special case of the TWP scheme wherein the sleep happens with regularity and is not dependent on the traffic flow. This mode is similar to IEEE 802.11 networks where the client's wireless card powers itself down and the Access Point buffers the packets [12]. While the Time Window Prediction scheme, in the presence of an accurate prediction function, does not necessarily increase latencies, PSM is more aggressive and is naturally expected to cause an increase in latency. PSM is more of a policy decision about powering down ports.

```
Inputs:
    Packet Threshold for sleeping: $\tau_p$
    Size of the observation time window: $t_o$
    Size of the sleep time window: $t_s$
    Lower bound for sleeping: $\tau_s$
    Bound for increase in latency: $L$
Variables:
    $noSleep$ = false
    Average long-term per-packet increase in
   latency: avg-lat = 0
Step 1: Count number of packets, num-packet, in
        $t_o$ window.
Step 2: If num-packet $< \tau_p$
            If noSleep is false
                Sleep for $t_s$ window
                Process buffered packets
                Calculate per-packet latency
                in the $(t_o + t_s)$ window, $delay_{recent}$
            Else
                Calculate per-packet latency
                in the $(t_o)$ window, $delay_{recent}$
            weighted-latency =
                $w_1*$ avg-lat + $w_2*$ $delay_{recent}$
            adapt-ratio = weighted-latency / L
            If ($t_s$ / adapt-ratio $> \tau_s$)
                $t_s \leftarrow t_s$ / adapt-ratio
            Else
                $noSleep$ = true
Step 3: Update avg-lat
Step 4: Go to Step 1
```

Table 2: **Time Window Prediction**

4 Performance Evaluation

We now present an initial evaluation of the TWP scheme. Results for PSM along with an exhaustive evaluation can be found in [10].

4.1 Evaluation Parameters

Our figures of merit are the percentage of power reduced as well as packet loss. Our baseline power consumption assumes all the switch's components to be powered up throughout. We calculate the reduction in $Power_{switch}$ because of our schemes. Prior work [17, 14] used the percentage of times when the port is in a low-power state as a metric for evaluation. But the overall power reduction is a better metric as powering down a fraction of ports does not reduce the power consumption of other components like the switching fabric and line-cards.

Traces: We evaluate traces from a Fortune 500 company's enterprise network of PC clients and file and other servers. Our enterprise traces were collected in the Fortune 500 company's LAN in March 2008 for a period of 7 days. We collected SNMP MIB counter data of the number of packets across every port (ingress and egress measured separately) on a switch with four line cards (192 ports). This counter data was collected once every 20 seconds. Consistent with previous studies [18], we assume a Pareto distribution of packets within the 20 second interval. This captures the bursty nature of traffic.

We put our results in context by comparing them with an optimal power reduction scheme that assumes an *oracle* to exactly predict each idle period. It also assumes an instantaneous transitioning between the power states of the switch. For our traces, the optimal power saving is 33.9% implying an utilization of 16.8%.

4.2 Results

Cluster Size: A cluster of ports are associated to a shadow port for receiving ingress packets when in low-powered state. Higher number of ports in a cluster will result in a higher probability of multiple ports in the cluster being in low-powered state at the same time, and hence savings in power. But higher cluster sizes also result in packet losses. Our experiments indicate a cluster size of 12 to be best.

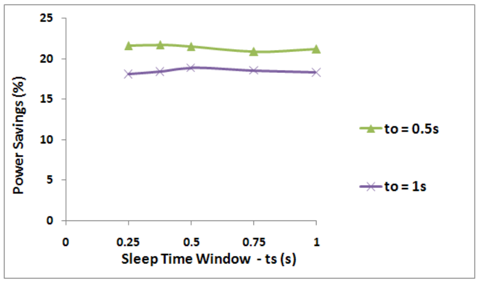

Figure 2: **Power Savings with TWP is independent of the sleep time window (t_s).**

Adaptation of t_s: TWP automatically adapts its sleep window, t_s, to meet the latency bounds. As shown in Figure 2, the power savings is a function of only t_o. For a fixed value of t_o, we experimented with varying initial values of t_s – 0.25s, 0.375s, 0.5s, 0.75s and 1s. The results illustrate the adaptive nature of the algorithm whereby the initial value of t_s is automatically and continuously modified to meet the latency bounds.

Wake on Packet: Table 3 illustrates the benefits of the Wake-on-Packet capability. Note that the power savings achieved with the Wake-on-Packet facility is 80% of the optimal power savings. While we have evaluated our schemes with a transition time, δ, of 10 ms, our results show that if improvements in hardware facilitate a 1ms transition, our savings are 90% of the optimal value.

t_o	Adaptive	WoP	Increase
0.5s	21.6%	27.3%	27%
1s	18.1%	24.4%	34%

Table 3: **Wake-on-Packet produces a significant increase in power savings in the TWP scheme**

Packet Loss: Figures 3 plots the packet loss for varying buffer sizes, for the adaptive TWP scheme and how it decreases under Wake-on-Packet. Packet-losses decay exponentially as the buffer size increases. During prolonged idle periods, the adaptive scheme automatically increases its sleep window. This is likely to cause packet losses when packet flow resumes at higher rates because shrinking the time window takes time. But with WoP, the sleep window remains constant during the indefinite sleep. The curves show a packet-loss of under 1% for buffer sizes greater than 500 KB. Most modern switches support such buffer sizes [1] and thus the TWP scheme produces acceptable packet-losses.

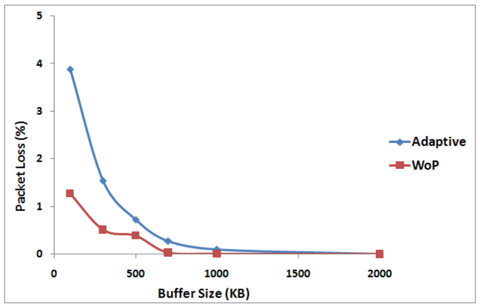

Figure 3: **Packet Loss with TWP**

5 Lightweight Alternative

We propose an alternate that addresses over-provisioning in network designs. In contrast to our earlier schemes, this one considers the macroscopic switch traffic. Also, the TWP and PSM schemes affect only the power consumed by the ports, bounding the amount of savings possible. As observed in prior work [16, 8], traffic patterns have a clear diurnal variation. The traffic resembles a sine curve that peaks in the day and experiences a lull in the nights. For instance, enterprise networks can expect to have far fewer users at 3AM compared to 3PM.

Our proposal is to deploy low-power or *lightweight* alternatives for every high-powered switch. The high-powered switches support very high packet processing speeds (in the order of Mega packets per second) and have multiple line cards with each of the cards connecting up to 96 machines through its ports. The lightweight alternatives are low-powered integrated switches with lower packet processing speed and line speeds. All machines have connectivity through the high-powered switch as well as the lightweight alternative. From the traffic patterns, the system automatically identifies ”slots” of low activity and ensure that only one of the two connections is appropriately powered-up and used depending on the traffic load. Recent work [19] on transferring state − routing tables and other configuration information − between routers can be employed for live transfer of state between the switch alternatives.

Identifying slots of low-activity can be done using K-Means clustering. A day is split into equal-sized slots and the number of packets per slot is logged for t training days. Every day's data is classified using K-Means clustering (K = 2), to produce two clusters: one each for high and low activities. The clustered data is processed to find the count of the total number of days when a particular slot is in the low-activity cluster. If the low-activity fraction of days is higher than a confidence level C, then that slot is marked as low-activity. All low-activity slots are served using the lightweight alternative.

Our initial evaluation demonstrates power savings of 15 to 32% for varying confidence levels. For detailed performance results, please refer to [10].

A 30% power reduction translates to an economic saving of $37,133 per year (10 cents/kWh). The economic benefits are clearly higher than the price of the lightweight alternatives and hence the schemes ensure that cost of the extra hardware is amortized.

6 Related Work

The IEEE 802.11b specification [12] includes access points packet-buffering schemes so clients can sleep for short intervals. This is similar to our Power-Save-Mode for switch ports. We augment this idea by incorporating a dynamic and automatic sleep period to bound latency.

Gupta et al. [13] identified the Internet's high power consumption and devised low-power modes for switches in a campus LAN environment [15]. Our Time Window Prediction scheme takes better advantage of extended idle periods and does not require the port to be on throughout the idle period. This advantage is significant when the traffic patterns are bursty with long idle periods. Also, we introduce latency bounds and investigate its effect on latency and packet loss.

Intelligent scaling of switch link speeds, depending on network flow, has been proposed [11, 3, 17]. An important practical problem is that the speeds on the switch are discreet (10 Gbps, 1 Gbps, 100 Mbps and 10 Mbps) and hence taking advantage of this automatic scaling would require vast differences in the traffic flows [17, 11]. Automatic scaling of link speeds also incurs the overhead of auto-negotiation of link speeds between the endpoints.

Nedevschi et al. [17] also talk about a ”buffer-and-burst” (B&B) scheme where the edge routers shape the traffic into small bursts and transmit packet in bunches so that the routers in the network can sleep. This is not applicable for traffic originating from the internal nodes and is also not incrementally deployable as it requires network-wide coordination.

7 Conclusion

We proposed power reduction schemes that focus on opportunistic sleeping and lightweight alternatives during idle or low-activity periods. The results of our schemes − power savings and performance − are encouraging. The advantage offered by smart hardware features like Wake-on-Packet and shadow ports leads us to recommend them in future switch designs.

References

[1] Cisco Catalog. *http://cisco.com/en/US/prod/collateral/switches/ps5718/ps708/product_data_sheet/0900aecd8017376e.html.*

[2] Cisco Power Calculator. *http://tools.cisco.com/cpc/launch.jsp.*

[3] Energy Efficient Ethernet. *http://www.ieee802.org/802tutorials/july07/IEEE-tutorial-energy-efficient-ethernet.pdf.*

[4] IEEE P802.3az Energy Efficient Ethernet Task Force. *http://www.ieee802.org/3/az/index.html.*

[5] Ipmon Sprint. The Applied Research Group. http://ipmon.sprint.com.

[6] AMD Magic Packet Technology, 2004. *http://www.amd.com/us-en/ConnectivitySolutions/TechnicalResources/0,,50_2334_2481,00.html.*

[7] Energy Efficient Ethernet, Outstanding Questions, 2007.

[8] J. S. Chase et al. Managing Energy and Server Resources in Hosting Centers. In *SOSP 01*, Oct 2001.

[9] Amit Jardosh et al. Towards an energy-star wlan infrastructure. In *HOT-MOBILE 2007*, Feb 2007.

[10] G. Ananthanarayanan and R. H. Katz. Greening the switch. Technical Report UCB/EECS-2008-114, EECS Department, University of California, Berkeley, Sep 2008.

[11] C. Gunaratne et al. Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed. In *International Journal of Network Management*, 2005.

[12] IEEE802.11b/D3.0. Wireless lan medium access control (mac) and physical (phy) layer specification: High speed physical layer extensions in the 2.4 ghz band. 1999.

[13] M. Gupta and S. Singh. The Greening of the Internet. In *SIGCOMM*, 2003.

[14] M. Gupta and S. Singh. Using Low-power Modes for Energy Conservation in Ethernet LANs. In *IEEE INFOCOM (Minisymposium)*, May 2007.

[15] M.. Gupta et al. A Feasibility Study for Power Management in LAN Switches. In *12th IEEE ICNP*, Oct 2004.

[16] Martin Arlitt and Tai Jin. Workload characterization of the 1998 world cup web site. In *HPL-1999-35R1, HP Laboratories,*, Sep 1999.

[17] Sergiu Nedevschi et al. Reducing network energy consumption via sleeping and rate-adaptation. In *NSDI '08*, Apr 2008.

[18] Srikanth Kandula et al. Walking the tightrope: Responsive yet stable traffic engineering. In *ACM SIGCOMM*, Aug 2005.

[19] Yi Wang et al. Virtual routers on the move: Live router migration as a network-management primitive. In *ACM SIGCOMM*, Aug 2008.