INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2005; **15**: 297–310
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nem.565

Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed

By Chamara Gunaratne, Ken Christensen*,† and Bruce Nordman

The IT equipment comprising the Internet in the USA uses about $6 billion of electricity every year. Much of this electricity use is wasted on idle, but fully powered-up, desktop PCs and network links. We show how to recover a large portion of the wasted electricity with improved power management methods that are focused on network issues. Copyright © 2005 John Wiley & Sons, Ltd.

1. Introduction

A growing expense and impact of the Internet is its energy use. Current estimates are that 2% of electricity consumption in the USA goes to powering the Internet.[1] In Germany it is estimated that energy consumption by IT equipment will be between 2% and 5% in 2010.[2] The 2% estimate for the USA totals more than 74 TWh/year or $6 billion per year. It is predicted that energy use of IT equipment is growing faster than energy use of any other type within buildings.[3] Much of this energy use is wasted. Energy use by IT equipment is not proportional to utilization of the equipment. A recent study by Lawrence Berkeley National Laboratory (LBNL) showed that 60% of all desktop PCs in commercial buildings remain fully powered-on during nights and weekends[4] with existing power management almost always disabled. Beyond the PC are the Ethernet link and workgroup switch. At present, these energy consumers have almost no means of power management. Existing Internet protocols including discovery and routing are also 'energy unaware'; future protocols need to be made energy aware. For existing protocols that cannot be changed, methods of accommodating current operation must be developed. In previous work we have shown that there exists the potential for savings of *billions of dollars per year* in the USA alone.[5–7] These savings are summarized in Section 6 of this paper. Energy costs are a part of the total cost of ownership of an IT operation. Savings in these costs are of interest to IT managers and companies are beginning to respond with network management products (such as Verdiem with its centralized power management controller[8]) to address this need.

An efficient device consumes energy proportional to its output or utility. Thus, an idle or lightly utilized PC or Ethernet link should not consume the same energy as one that is highly utilized. In this paper, we develop several new methods to reduce energy consumption of PCs,

Chamara Gunaratne is a graduate student in the Department of Computer Science and Engineering at the University of South Florida.

Ken Christensen is an Associate Professor in the Department of Computer Science and Engineering at the University of South Florida.

Bruce Nordman is a Principal Research Associate in the Energy Analysis Department, Environmental Energy Technologies Division, of Lawrence Berkeley National Laboratory Berkeley, California.

**Correspondence to: Ken Christensen, Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA*
†E-mail: christen@csee.usf.edu

Ethernet links, and first-level LAN switches. We also explore methods for reducing energy consumption on a larger scale by (1) supporting centralized proxying and control for discovery protocols, and (2) disabling unused paths in the scope of routing. Key to any method of power management is that it be invisible to the user. Performance impacts of recovery from system wake-up and of reduced levels of operation must be understood. These are significant performance evaluation problems that are part of what we believe is the 'next frontier'.[5] New ideas are needed to (1) increase the number of desktop PCs that have power management enabled and functioning, and (2) increase the energy efficiency of actively functioning network components. We address both of these areas in this paper.

The remainder of this paper is organized as follows. In Section 2 we describe power management and existing wake-up mechanisms for PCs. In Section 3 we cover new ideas in proxying and application-specific wake-up to enable power management in PCs. Section 4 describes the problem of persistent TCP connections and how to split a connection to enable power management. Section 5 addresses scaling of link and switch data rates to utilization. Section 6 quantifies the expected energy savings achievable from the methods described in this paper. In Section 7 related work is covered. Finally, Section 8 is a summary.

Current estimates are that 2% of electricity consumption in the USA goes to powering the Internet.

2. Power Management for Reducing Energy Consumption

Electricity consumption of IT equipment has been a research focus of the Environmental Energy Technologies Division of the Lawrence Berkeley National Laboratory (LBNL) for over 15 years.[9] Their studies have documented that approximately 2% of total USA electricity production goes to powering IT equipment. To address this large electricity use, the Energy Star program administered by the US Environmental Protection Agency (EPA) and the Department of Energy (DOE) mandates energy efficiency requirements for equipment purchased by US government agencies. These standards have driven both manufacturer offerings and consumer purchasing behavior and have become widely adopted worldwide. The Computer Specification (Version 3.0) for Energy Star[10] specifies that the ability of a PC to go to sleep should not be compromised by attaching it to a network and that it have "the ability to respond to wake events directed or targeted to the computer while on a network"—this is essentially what an Ethernet NIC with Wake On LAN (WOL) can do. WOL is described in Section 2.2. It is estimated that the Energy Star office equipment program will save at least $30 billion between 2001 and 2010 with about two-thirds of this saving coming from PCs and monitors.[11] Energy Star does not address what are wake events or issues with waking-up too often or not enough. The capabilities specified by Energy Star are usually disabled as documented in Reference 4; we believe that the failure to maintain network connectivity in sleep is the principal reason. These are open problems that we address.

Power management features are often disabled due to the inconveniences that they present to users and network administrators.

2.1. Power Management in PCs

To improve the battery life of mobile computers and to meet Energy Star specifications, PC hardware and software implement power management. The Advanced Configuration and Power Interface (ACPI) specifies power management concepts and interfaces.[12] ACPI integrates the operating system, device drivers, system hardware components, and applications for power management. ACPI defines several power states for each component ranging from fully powered-on to fully powered-off with each successive state consuming the same or less power. Our measurements found that a Dell 2350 1.8 GHz Pentium 4

PC (system unit only) consumes 6W in its sleep state and 60 to 85W when fully powered on.[5] Studies done by LBNL have shown that power management capabilities are largely disabled. In Reference 4 it was found that of 1453 desktop PCs in 12 non-residential buildings, 36% were turned off, 60% were on and only 4% had power management functioning and so were asleep.

Power management features are often disabled due to the inconveniences that they present to users and network administrators. For example, network administrators often need night-time access to PCs for applying software patches, performing back-ups, and so on. One approach to address these inconveniences is global control of power management settings in the PCs of an enterprise. Verdiem markets a product to estimate energy use and control power management settings.[8] The product consists of a global manager and client programs run on each managed PC, for which a minimum energy saving of $20 per PC per year is claimed. A global approach to power management cannot solve open problems with keeping TCP connections alive between clients and servers and also cannot respond to short-term changes.

2.2. Wake On LAN (WOL) for PC Wake-up

In 1995 Advanced Micro Devices (AMD) developed a packet-based method, called Magic Packet, for waking-up a networked PC through its network interface controller (NIC).[13] Magic Packet is also known as Wake On LAN (WOL). A WOL packet contains the MAC address of the receiving NIC 16 times. On receiving such a packet, a WOL NIC will interrupt the PC to begin a wake-up sequence. A WOL NIC is powered at all times through an auxiliary power connection, even when the rest of the system is asleep. A WOL NIC is shown in Figure 1 where the WOL cable contains power and interrupt lines that are plugged-in to the PC motherboard. The ACPI standard defines an expanded set of triggers for the network interface that would cause the NIC to trigger a wake-up of the PC. These include:

- A traditional WOL packet
- A packet directed at the PC's IP address
- Operating system specified wake-up triggers

WOL is limited in scope by the need to know the MAC address of a NIC. Such knowledge of a MAC address, and the need to know that the system is asleep and needs to be sent a wake-up packet, does not exist in an IP routed network. Directed packet wake-up solves this problem at the expense of causing wake-ups when not needed.

3. Protocol Proxying and Improved Wake-up

A shortcoming with existing wake-up based schemes is wake-up for useless or minor tasks. It

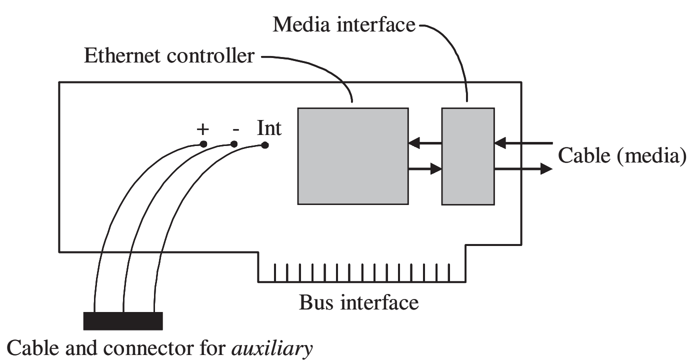

Figure 1. Ethernet NIC card with auxiliary power and wake-up capability

should be possible to be more selective in wake-up and to proxy for some tasks normally performed by the PC operating system (which requires the PC to be fully powered-on). We explore the possibilities of improved wake-up and proxying for responding to 'network chatter'. An idle PC receives packets on its link at all times. The packets can be categorized as:

- *No response required.* Packets that require no actions or response and are discarded by the protocol stack in the operating system. This includes broadcast bridging and routing protocol packets. This also includes 'hacker' traffic such as port scans.
- *Minimal response required.* Packets that require minimal action or a response that a proxy could handle. This includes ARP and ping packets.
- *Wake-up required.* Packets that require operating system or application-level response. This includes TCP SYN packets for connection requests to applications with listens and SNMP GET requests.

In addition, some network protocols generate packets from a client. For example, DHCP lease renewal requests are generated locally from a PC or other device holding a DHCP granted IP address. Packets of the first two types (no response required and minimal response required) we call 'network chatter'. Table 1 summarizes 296 387 packets received in 12 hours and 40 minutes by an idle PC connected to the University of South

Protocol	% in trace	Discard	Proxy	Wake-up	Comments
ARP Request and Reply	52.50 %	Most	Yes	No	Requests for IP address answered by proxy
Universal Plug & Play	16.50	Most	Yes	No	Discovery messages answered by proxy
BRIDGE "Hello"	7.80	All	No	No	Can be discarded by proxy
Cisco Discovery	6.90	All	No	No	Can be discarded by proxy
NetBIOS Datagram	4.40	Some	No	Possible	Directed packets need a PC wake-up
NetBIOS Name Service	3.60	Some	Yes	Possible	Directed packets need a PC wake-up
Banyan System	1.80	All	No	No	Can be discarded by proxy
OSPF	1.60	All	No	No	Can be discarded by proxy
DHCP	1.20	All	No	No	Can be discarded by proxy
IP Multicasts	1.00	All	No	No	Can be discarded by proxy
RIP	0.50	All	No	No	Can be discarded by proxy
SMB	0.40	Some	No	Possible	Directed packets needs a PC wake-up
NetBEUI	0.31	All	No	No	Can be discarded (deprecated protocol)
Unknown port scans	0.30	All	No	No	Contains TCP SYNs (unneeded wakeups)
BOOTP	0.25	All	No	No	Can be discarded by proxy
NTP	0.20	All	No	No	Can be discarded by proxy
NetBIOS Session service	0.12	Some	No	Possible	Directed packets need a PC wake-up
ICMP (including ping)	0.08	Some	Yes	No	Echo requests answered by the proxy
DEC Remote Console	0.08	All	No	No	Can be discarded by proxy
SNMP	0.06	Some	No	Possible	Can be discarded unless SNMP agent
ISAKMP	0.04	All	No	No	Can be discarded by proxy
X Display Manager	0.02	All	No	No	Can be discarded by proxy

Table 1. Breakdown of received packets on an idle PC connected to the USF network

Florida network—over 6 packets per second. ARP packets constitute the slight majority of received packets with Universal Plug and Play, routing, and bridge protocol packets comprising about 30% of the total. The packets summarized in Table 1 were traced using a protocol trace tool running on another PC on a shared Ethernet repeater. The NIC was configured to wake-up the PC upon receiving either a WOL packet or a directed packet. The system was configured for 10 minutes of inactivity time before transitioning into the Windows XP 'standby' (low-power sleep) state. PCs and most other electrical devices consume more energy during the wake-up transition—a wake-up power spike—than during steady state operation. Thus, unnecessary wake-ups waste energy in two ways. Energy is wasted by the wake-up power spike and during the steady-state on time before re-entering the sleep state.

3.1. Proxying to Reduce the Effect of Chatter

Protocol proxying implemented on a NIC or within a first-level LAN switch could handle most network chatter and eliminate the need to wake-up the full system for trivial (network related) tasks. A NIC could include a small processor and software that would, when the PC is in a low-power state, act as proxy for the PC protocol stack and applications. The proxy would filter packets that require no response, reply to packets that require a minimal response, and only wake-up the system for packets requiring a non-trivial response. The proxy would also generate packets for simple protocols such as DHCP based on a timer interrupt. This response to an internal event is similar in complexity to a minimal response to an external event such as a received packet. Interfaces need to be defined to pass state information from a PC to its proxy in a NIC or LAN switch. This state information would include the IP address and which TCP ports are open for listening.

With proxying, fully 91% of the nearly 300 000 packets summarized in Table 1 would be filtered-out or trivially responded to by a proxy. The remaining 9% of packets included TCP SYN packets, most for non-existent ports. The incoming ARP requests and ICMP packets can be responded to by a proxy. The SYN packets require a response from the system only when there is an executing application with an open port.

3.2. Improved Wake-up Semantics

Existing wake-up methods rely on a special format packet (the WOL packet described in Section 2.2), trigger on the appearance of the local IP address in an ARP packet, or match and trigger on pre-programmed patterns found in a received packet.[14] Direct wake-up often results in many unnecessary wake-ups with the system being powered-up when it need not be. A more selective and 'intelligent' wake-up would reduce the amount of time a system is awake by preventing spurious wake-ups. The traced PC of Table 1 was awake for 2 hours and 40 minutes (21%) of the 12 hour 40 minute trace period due to wake-ups caused by incoming TCP SYN packets (after each wake-up the PC would stay awake for approximately 2 minutes). However, apart from Windows workgroup networking ports, there were no open TCP ports (application-level listens). Of the wake-up causing TCP SYN packets recorded in the trace, 53% were for open local area networking ports. Adding knowledge of the open TCP port numbers to triggering wake-ups on TCP SYN packets would reduce the observed system's powered-up time to 1 hour 24 minutes (11%). Open TCP ports are created by network applications when they issue a listen command for incoming TCP connection requests.

An external proxying capability (e.g., in the first-level switch) would generate WOL wake-ups for its proxied systems on detection of a wake-up event. This is described in Reference 15. Placing the proxy and wake-up capability externally would require passing state information between the PC and its proxy, but would not require hardware changes in PCs since all modern Ethernet NICs support WOL. Both approaches (proxying on the NIC and LAN switch) could be employed.

4. Managing TCP Connections

Many applications maintain a permanent TCP connection between a client and server. For example, a client/server database application may leave TCP connections open at all times. Telnet, SSH, and other shared resource applications also require permanent connections. Even when the TCP connection is idle—that is, no application data is flowing on the connection—both end-points must generate and respond to periodic TCP-generated keep-alive messages that occur at least once every 2 hours. Many applications also generate keep-alive messages that must be responded to if the application is to maintain its TCP connection. Thus, the problem is how to respond to keep-alive messages without requiring the full resources of a client or server. In addition, there must be a means to quickly and fully resume the TCP connection when application data needs to flow either from client to server or server to client. There are three possible ways to address persistent TCP connections:

1. Rewrite applications at the client and/or server that normally maintain persistent TCP connections to only be connected while actively requiring data transfer. HTTP 1.0 uses this approach.
2. Use proxying at the client to answer keep-alive messages and wake-up the client PC on resumption of data transfer within a connection
3. Split the TCP connection such that applications 'see' a connection, but TCP has closed the connection.

The first method is outside the scope of this paper. The second method based on proxying would require no changes in the server. The widely used method for sending TCP keep-alive messages is to send packets with connection sequence numbers that have already been acknowledged by the receiver. Upon receiving these already acknowledged sequence numbers, the receiver should transmit a duplicate acknowledgement for the last data received from the sender. Implementing this in a proxy would require the proxy to be aware of the connection sequence numbers and acknowledgements for every established TCP connection. This would be non-trivial for a proxy to implement. The third method based on splitting a TCP connection within the client and server would require additional software in both the server and client, but no changes to TCP or to the applications. We explored this third method.

4.1. A Split TCP Connection

To avoid changes to applications and the TCP protocol implementation, we split a TCP connection with the addition of a 'shim' layer between the sockets interface and the application. This shim presents a sockets interface to the application (thus, the application does not change) and uses the existing sockets layer of the TCP software implementation. Today, when a client powers-down, the connection is dropped and the server cleans-up all resources (state) associated with the connection. The application is then notified that the connection is closed resulting in an error if the application expects a persistent connection. Our shim layer 'fakes out' applications to see an established connection at all times. The shim does the following:

- When the client at one end of the connection transits to low-power sleep mode, the shim layer informs the opposing shim layer in the server to drop the TCP connection.
- When the client transits back to fully powered-on (e.g., due to keyboard or mouse activity), the shim layer re-establishes the previously dropped TCP connection by informing the opposite shim layer which socket connection to connect to.
- If a server application wishes to send data to a client that is in low-power sleep state, the shim layer first wakes-up the sleeping client (e.g., using one of the wake-up mechanisms already described) and then re-establishes the TCP connection.

Figure 2 shows the semantics of split TCP connection. It can be seen that the shim layer's functionality is used when power management events are triggered.

To evaluate the feasibility of this method, we implemented a prototype by using a telnet client and a telnet server running on Windows XP. Since one cannot modify the Windows socket library source code, the shim layer was implemented in the application space and compiled in with the application source code. It was also not easily possible for us to trap and respond to the

	10 Mb/s		100 Mb/s		1000 Mb/s (1 Gb/s)	
	On	Sleep	On	Sleep	On	Sleep
Intel PRO 1000/MT (on motherboard)	57.9 W	3.2 W	58.2 W	3.6 W	60.6 W	7.0 W
NetGear GA311 (PCI)	59.6	5.3	59.6	5.7	62.9	5.8
LinkSys EG1032 (PCI)	59.1	5.6	59.6	5.5	62.0	5.6
Change from 10 Mb/s (On = 58.9 W, Sleep = 4.7 W)	—	—	0.3	0.2	2.9	1.4

Table 2. Power consumption of a Dell GX270 Pentium 4 PC with different NICs

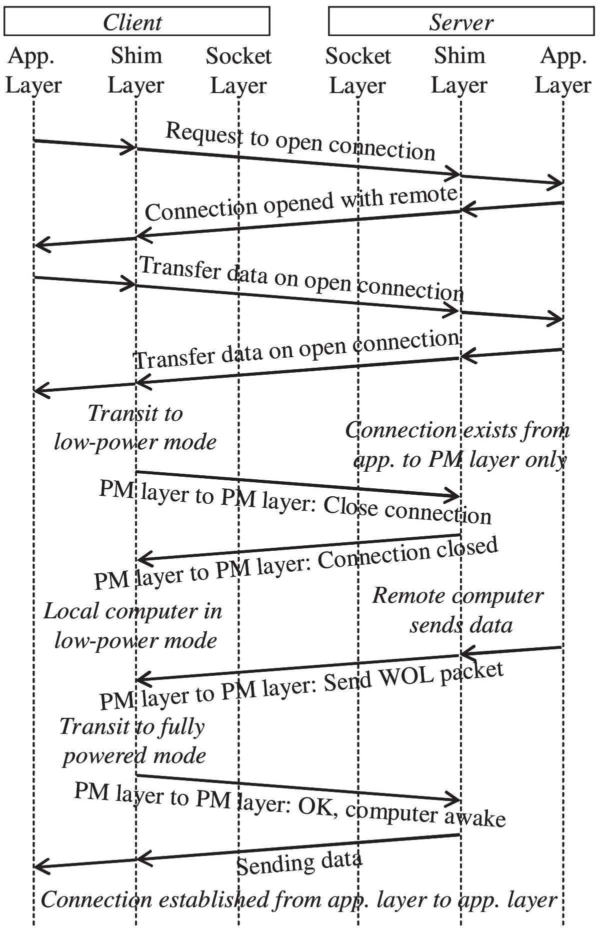

Figure 2. Semantics of a split TCP connection

Windows `WM_POWERBROADCAST` power management event notifications. Therefore, to test the prototype shim we used manual event triggers. It was possible to drop and re-establish the underlying TCP connection without losing the telnet session between client and server.

5. Scaling of Link and Switch Speed

Most PCs are connected to the Internet via an Ethernet link to a first-level or workgroup LAN switch. The standard data rate of this link has increased from 10 Mb/s to 1 Gb/s. Ethernet at 10 Gb/s is already available for fiber and an IEEE study group is working on standardizing 10 Gb/s Ethernet for unshielded twisted pair (UTP) cabling. The final standard is expected in 2006 and anticipated to be inter-operable with existing 10/100/1000 Mb/s UTP Ethernet standards. This could lead to the widespread deployment and commoditization of 10 Gb/s Ethernet NICs.

The power consumption of Ethernet NICs and switches was measured for a range of data rates and utilization levels. Table 2 shows the power consumption of an idle Dell GX270 Pentium 4 PC (system unit only) with different NICs installed. The Intel PRO 1000 NIC is available on the motherboard and when other NICs were installed was disabled using the BIOS. The measurements were taken using an AC power meter at the wall socket and show that there is an approximately 3 W difference in power consumption between running at 10 Mb/s and 1 Gb/s data rate (with no traffic). When transitioning to Windows XP standby mode, the NIC automatically drops the data rate from 1 Gb/s to 10 Mb/s. Preliminary measurements were made of a fiber optic 10 Gb/s Ethernet NIC from S2io. Results show that this NIC increases power consumption by 18 W while powered-on.

Figure 3 shows power consumption measurements of a Cisco Catalyst 2970 switch. Power consumption of the switch is plotted with respect to the number of active links for various Ethernet data rates. The results show that when increasing

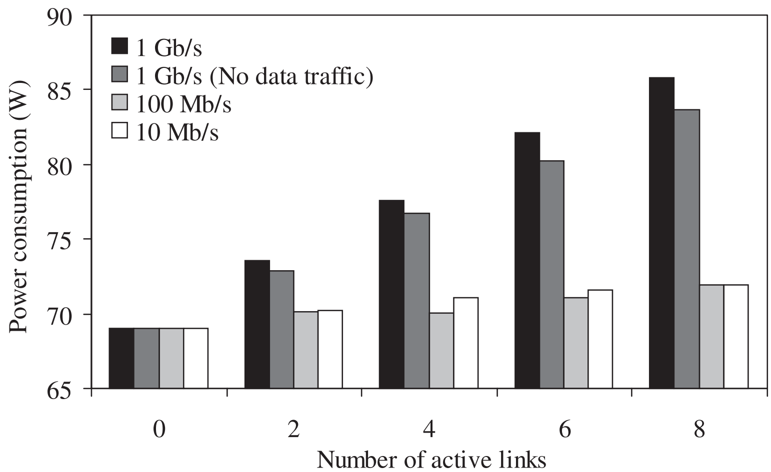

Figure 3. LAN switch power consumption

the number of 10 Mb/s and 100 Mb/s links, the switch power consumption increases by a negligible amount (0.3 W per link). When increasing the number of 1 Gb/s Ethernet links, the power consumption of the switch increases by approximately 1.8 W per link. From Table 2 and Figure 3 it is seen that operating Gigabit Ethernet NICs at 10 Mb/s or 100 Mb/s data rate (instead of 1 Gb/s data rate) can yield a saving of approximately 4 W (2.7 W at the PC and 1.5 W at the switch) per link. From this we ask two questions:

1. What percentage of time would a lower link data rate yield the same user performance (measured in user-perceived delay) as would a higher data rate?
2. Is it possible to switch between link data rates in real time (i.e., while the link is active) and gain an energy saving at no perceivable increase in delay?

5.1 Link Data Rate as a Function of Queue Length

We propose to scale Ethernet link data rate as a function of queue length in both the PC and LAN switch. The possible power savings depend upon the link utilization of PC users and the effectiveness of the algorithms used for scaling the data rate to take advantage of periods of low utilization. The queue length (e.g., of the TCP send buffer in a PC or of the buffer in a switch line card) and the link utilization level are used as inputs to an algorithm (see Figure 4) to dynamically change the link data rate. When the queue length is greater than the high queue threshold, the data rate is set to the high data rate. When the queue length is less than the low queue threshold and if the link utilization is also less than the link utilization threshold the data rate is set to the low data rate. Threshold values are described later.

```
if (link data rate is high)
  if (queue length is less than low queue threshold)
    if (link utilization is less than link utilization threshold)
      set the link data rate to low
else if (link data rate is low)
  if (queue length is greater than high queue threshold)
    set the link data rate to high
```

Figure 4. Algorithm for scaling Ethernet link data rate

Ethernet NICs capable of operating at multiple data rates use a mechanism called Auto-Negotiation to determine at which data rate to operate. When devices Auto-Negotiate they exchange a series of code words embedded in the standard Ethernet link pulse that describe the data rates they are capable of supporting and establish the link data rate at the highest rate common to both devices. To maintain compatibility with 10BASE-T, Auto-Negotiation code words are exchanged every 16 milliseconds. For 1 Gb/s Ethernet devices, completing this process will take a minimum of 256 milliseconds. Using the existing Auto-Negotiation capability to change the data rate dynamically would disable the link for hundreds of milliseconds. This would result in unacceptable levels of packet loss and/or require unreasonably large buffers. Optical Gigabit Ethernet transceivers are capable of link acquisition in tens of microseconds. A link start-up time for interconnection networks in 10 000 clock cycles is possible.[16] We thus estimate that it would be possible for copper Ethernet link data rates to be dynamically changed in 1 to 5 milliseconds. We use switching times of 1, 3, and 5 milliseconds in our simulation study. A MAC frame based approach—similar to the IEEE 802.3 PAUSE MAC frame for flow control—can be adopted to signal opposite ends of a link for a desired data rate.

To analyze the effects of data rate scaling and to identify possible energy savings, a simulation model was developed that scales between 10 Mb/s and 100 Mb/s. The input to the simulation model is a packet trace containing tuples of packet interarrival time and packet length. We used a 30 minute duration packet trace of the user with the largest data transfers in the USF campus network (this is a dormitory PC that is very likely being used for file sharing). In 30 minutes this extreme user transferred 901 MB of data. The mean packet size was 577 bytes and the mean packet interarrival time was 1.1 milliseconds. This user is connected to the USF gigabit backbone by a 100 Mb/s Ethernet link. Thus, in our simulation we studied changing between a low link data rate (e.g., 10 Mb/s) and 100 Mb/s. In the simulation model packets were queued before transmission. Simulation experiments were run with maximum queue (buffer) capacities of 250, 300, 350 and 400 packets. At the mean packet size, this would roughly correspond to buffer capacities of about 140, 170, 196 and 225 kB. The low queue threshold was set to 20% of the maximum queue capacity and the high queue threshold was set to 40%. The link utilization threshold was set to 3 Mb/s (as measured in the previous one second) and the low and high data rates were set to 10 Mb/s and 100 Mb/s. Time spent changing from one data rate to the other was regarded as time spent in the higher data rate.

5.2 Simulation Results

Figure 5 shows the percentage of time spent in low data rate and the mean latency for the four different queue capacities and three switching times. For a given queue capacity, the mean latency values were quite close and therefore a single rounded-up value is shown. For the lowest queue size, it is possible to maintain the low data rate for 35% of the trace time with a mean latency of 2 ms. Figure 6 gives the percentage of time spent in the low data rate for differing low data rates. In these simulations we fixed the high data rate at

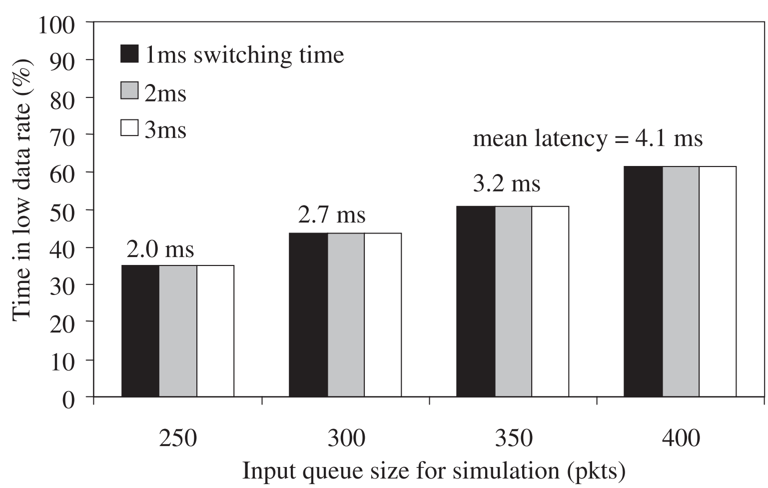

Figure 5. Time spent in low data rate (mean latency is shown)

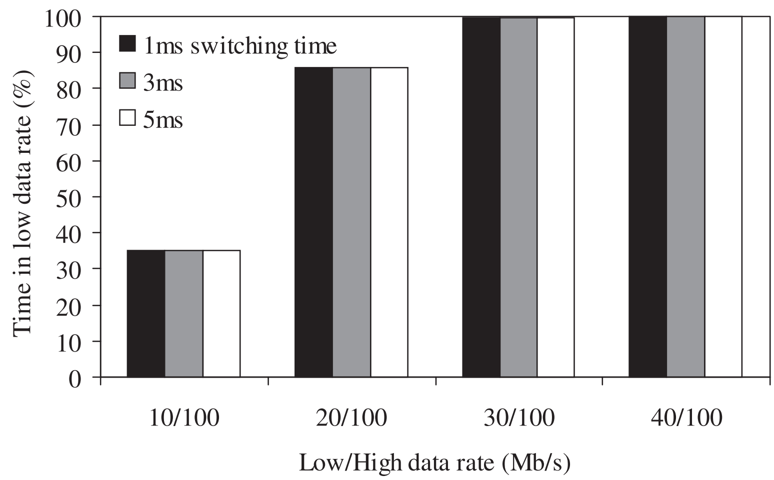

Figure 6. Time spent in low data rate as a function of low/high link rates

100 Mb/s and vary the possible low data rate from 10 Mb/s to 40 Mb/s. There are no existing 20, 30, or 40 Mb/s Ethernet links; we used this hypothetical link data rate only to see the effects on performance if such a link data rate did exist. The queue capacity is fixed at 250 packets. The results show that at lower data rates of 30 Mb/s and 40 Mb/s almost no time is spent at 100 Mb/s.

These results show that even for the busiest user on the USF campus it is possible to operate his or her link at a lower data rate with no significant increase in delay.

These results show that even for the busiest user on the USF campus it is possible to operate his or her link at a lower data rate (in this case, 10 Mb/s instead of the maximum possible 100 Mb/s) with no significant increase in delay. Given these results for an extreme user, it is plausible that when 1 Gb/s connections are brought to users, operation at 10 or 100 Mb/s for a large amount of time will be possible with no noticeable effect on the users. Existing work suggests that network links typically operate at low utilizations and will continue to do so.[17] The possible energy savings from this are described in Section 6.

6. Quantifying the Expected Energy Savings

In this section we estimate the possible energy savings from the methods described in this paper. Just how much energy can be saved is highly uncertain and depends on how many product types are affected, how many of each are present in homes and offices, what portion of the stock incorporates the new technologies, and how much time products spend in each operating mode (as determined by how they are configured and how they are used). Given this, any estimate can be only indicative of the magnitude of potential savings, and so we have kept it simple, basing it on that from Reference 7, which takes PC populations from what is expected for 2007.

Key assumptions and results are shown in Table 3. We aggregate current PC usage time into an average PC with five portions of time: three on (high-traffic, low-traffic, possible-sleep), sleep, and off. Link data rate reduction affects low-traffic and possible-sleep; proxying affects only possible-sleep time. We distinguish between PCs powered-on continuously and those powered-on only sporadically. Commercial usage patterns are informed by Reference 4 and residential by Reference 18, though we extrapolate trends to 2007. From present trends we assume a population in 2007 of about 160 million desktop PCs in the combined residential and commercial sectors. We

Parameter		Value
PC power: On (W)		60
Sleep (W) w/ link data rate reduction		5
Savings: Sleep savings (W)		55
Link data rate reduction (W)		4
Average PC—operating time		
Off (%)		37.5
On—high-traffic plus low-traffic (%)		17.5
high-traffic (%)		3.5
low-traffic (%)		14.0
possible-sleep (%)		45.0
Savings	Link data rate	Proxying
% of year	59.0	45.0
Hours/year	5170	3940
kWh/year	21	217
$/year per PC	1.50	15.60
US savings ($ billion/year)	0.24	2.50

Table 3. Energy savings assumptions and potentials

assume that 50% of PCs (residential and commercial) are 'continuous on' machines that are potentially power managed and in a low-power sleep state 75% of the time. For the rest, we assume that they are on 20% of the time and potentially in a low-power sleep state for half of that. Savings calculations are unaffected by the amount of time spent in the sleep and off modes. Finally, we estimate that 80% of on-in-use time is 'low-traffic' time where a link data rate reduction could be achieved with no perceivable impact to the users. The link data rate reduction savings shown are both those for the PC and the additional savings at the LAN switch.

We assume an average electricity price of 7.2 cents/kWh. For the USA alone, the savings from link data rate reduction are about $240 million per year and the savings from proxying are about $2.5 billion per year. These savings are for the entire stock but are otherwise conservative for many reasons, particularly the exclusion of savings from notebook PCs and the extension of these technologies to non-PC devices such as consumer electronics. Energy savings from a reduction in cooling requirement are not considered here, but could be considerable in southern climates.

From the perspective of an organization that has many PCs, Figure 7 shows the projected savings for 100, 1000, and 10 000 PCs for different electricity prices (6, 8, and 10 cents per kWh). The saving per PC per year is the combined saving from link data rate and proxying from Table 3. For an organization with 10 000 PCs the savings per year ranges from about $140 000 to about $250 000.

7. Related Work

The first investigation of how PCs can be powered-off and network connectivity still be maintained was presented in Reference 19, where the design and evaluation of a new TCP connection sleep option in a 'Green TCP/IP' was described. This work was followed by an investigation of how a proxy server could allow multiple proxy-managed clients (e.g., desktop PCs) to remain in a sleep state for long periods of time.[15] In Reference 5 the idea of proxying on a NIC—this being infeasible in 1998 (when Reference 15 was published)—was proposed. In this paper, we evaluate the feasibility of proxying using a packet trace, which was not done in References 5 or 15. In Reference 5 we also describe the TCP split connection which improves upon the ideas in Reference 19 by not requiring changes to the TCP implementation in the operating system kernel. In

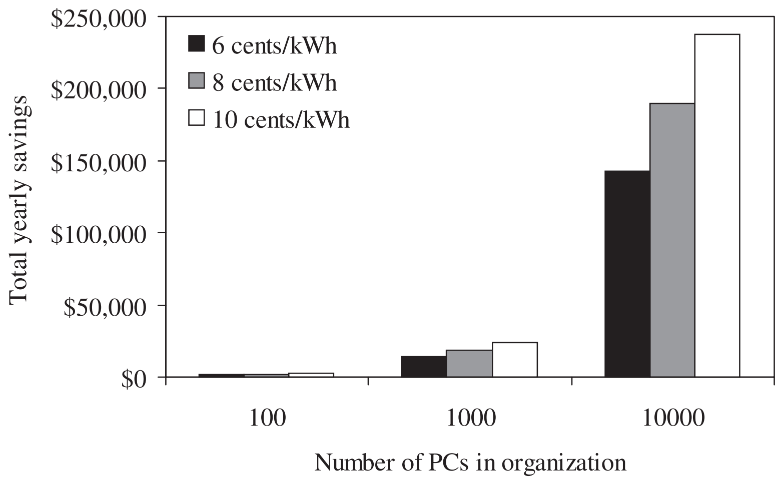

Figure 7. Total yearly savings for an organization with 100, 1000, and 10000 PCs

the context of mobile systems, there is existing work on connection resumption after failover[20] and migration of connections for mobile and distributed hosts[21,22] by using intermediate layers to isolate applications from the network protocol stack. Though conceptually similar, the split TCP connection method in this paper focuses on closing and resuming network connections without application awareness to avoid losing application state associated with a TCP connection.

A very significant work on energy consumption of the Internet was Gupta and Singh's ACM SIGCOM 2003 paper,[23] where it was calculated that in 2000 networking devices consumed about 6 TWh of electricity and that this value was expected to increase by another 1 TWh by 2005.[23] It is proposed that network interfaces in routers and switches can be put to sleep during idle times to reduce power consumption. Changes in routing protocols would need to be considered to achieve this. In Reference 24 power management capabilities for LAN switches are proposed. It is shown that a LAN switch that can enter a sleep state and be woken from it by packets queueing in a buffer (the buffer memory is not powered-off) can result in significant savings. In this paper, we show that a much simpler scaling of link data rate for NICs in desktop PCs and first-level LAN switches can achieve considerable energy savings.

The methods described in this paper if modestly adopted would result in savings of about $2.7 billion per year in the USA alone.

8. Summary and Future Work

We have developed and evaluated several new methods for addressing improved power management of IT equipment. Proxying, application-specific wake-up, and split TCP connections can all be used to increase the low-power sleep time of PCs. To implement these capabilities requires the addition of proxying capability to NICs and/or LAN switches. The added cost implementing proxying (e.g., for the addition of a small processor to a NIC) will be more than offset by the achieved electricity savings. To improve the energy efficiency of active (in use) PCs and their first-level LAN switch, link data rate reduction can be implemented. Scaling the data rate of an active Ethernet link as a function of user demand can result in significant savings with no perceivable impact to a user. The methods described in this paper if modestly adopted would result in savings of about $2.7 billion per year in the USA alone.

Future research areas include investigating user network traffic patterns and developing the necessary mechanisms for rapidly changing the Ethernet link data rate. Proxying and wake-up for wireless (IEEE 802.11) connections need to be further explored. In addition future work in investigating how discovery and routing protocols can better enable power management is needed. Many discovery protocols—such as the Simple Service Discovery Protocol (SSDP) used in Universal Plug and Play (UPnP)—are broadcast based. SSDP assumes that all devices on a network are listening at all times for discover messages and also periodically broadcasting advertise messages. Can a centralized proxy controller for SSDP reduce this need for end nodes to always be fully powered-on (to generate and respond to SSDP messages)? The possibility of changing routing protocols to disable paths with low utilization is described in Reference 23. Rather than changing routing protocols could a centralized route controller be used to disable and enable links as needed? Such an approach would eliminate the need to change existing routing protocols. Further work in energy-aware network protocols and applications needs to be pursued due to the potential for large economic and environmental savings.

Acknowledgements

The authors thank Adam Jacobs and John Wernicke, students at the University of Florida, for their assistance with the energy measurements presented in this paper.

References

1. Kawamoto K, Koomey J, Nordman B, Brown R, Piette M, Ting M, Meier A. Electricity used by office equipment and network equipment in the US: Detailed report and appendices. *Technical Report LBNL-45917*, Lawrence Berkeley National Laboratory, February 2001.
2. Thomas S, Barthel C. www.internet.co2, 2004. URL: http://library.iea.org/dbtw-wpd/Textbase/weo/papers/ictfeb02/THOMAS.PDF.
3. Roth K, Goldstein F, Kleinman J. Energy consumption by office telecommunications equipment in commercial buildings volume 1: Energy consumption baseline. *Arthur D. Little Reference No. 72895–00*, January 2002.
4. Roberson J, Webber C, McWhinney M, Brown R, Pinckard M, Busch J. After-hours power status of office equipment and inventory of miscellaneous plug-load equipment. *Technical Report LBNL-53729*, Lawrence Berkeley National Laboratory, January 2004.
5. Christensen K. The next frontier for communications networks: Power management. *Proceedings of SPIE—Performance and Control of Next-Generation Communications Networks*, Vol. 5244, pp. 1–4, September 2003.
6. Christensen K, Nordman B, Brown R. Power management in networked devices. *IEEE Computer* 2004;**37**(5):91–93.
7. Nordman B, Brown R. Networks: Tomorrow's big challenge for IT energy use. *Woodrow Wilson Center Workshop on Environment and the Information Economy: Next Steps for Research*, 15 March 2004.
8. Surveyor: The Network Energy Manager, Verdiem Corporation, 2004. URL: http://www.verdiem.com/surveyor.shtml.
9. Energy End-Use Forecasting Group, Lawrence Berkeley National Laboratory, 2004. URL: http://enduse.lbl.gov/.
10. Computer Memorandum of Understanding for Energy Star, 2004. URL: http://www.energystar.gov/ia/partners/product_specs/program_reqs/computer.mou.v3.pdf.
11. Webber C, Brown R, Mahajan A, Koomey J. Savings estimates for the Energy Star voluntary labeling program: 2001 status report, Lawrence Berkeley National Laboratory, *LBNL-48496*, February 2002.
12. Advanced Configuration and Power Interface Specification, Revision 2.0c, August 2003. URL: http://www.acpi.info/spec20c.htm.
13. AMD Magic Packet Technology, 2004. URL: http://www.amd.com/us-en/ConnectivitySolutions/TechnicalResources/0,,50_2334_2481,00. html.
14. Network Device Class, Version 2.0, Microsoft Corporation, 12 October 2000. URL: http://www.microsoft.com/whdc/resources/respec/specs/pmref/PMnetwork.mspx.
15. Christensen K, Gulledge F. Enabling power management for network-attached computers. *International Journal of Network Management* 1998;**8**:120–130.
16. Soteriou V, Peh L. Dynamic power management for power optimization of interconnection networks using on/off links. *Proceedings of the 11th Symposium on High Performance Interconnects*, pp. 15–20, August 2003.
17. Odlyzko A. Data networks are lightly utilized, and will stay that way. *Review of Network Economics* 2003;**2**(3):210–237.

18. Nordman B, Meier A. Energy consumption of home information technology. prepared for the U.S. Department of Energy, *LBNL-53500*, 2004.
19. Irish L, Christensen K. A 'Green TCP/IP' to reduce electricity consumed by computers. *Proceedings of IEEE Southeastcon*, pp. 302–305, April 1998.
20. Alvisi L, Bressoud T, El-Khashab A, Marzullo K, Zagorodnov D. Wrapping server-side TCP to mask connection failures. *Proceedings of INFOCOM*, pp. 329–337, April 2001.
21. Zandy V, Miller B. Reliable network connections. *Proceedings of the 8th ACM MobiCom*, pp. 95–106, September 2002.
22. Zhang Y, Dao S. A 'Persistent Connection' model for mobile and distributed systems. *Proceedings of the Fourth International Conference on Computer Communications and Networks*, pp. 300–307, September 1995.
23. Gupta M, Singh S. Greening of the Internet. *Proceedings of the ACM SIGCOMM*, pp. 19–26, August 2003. Revised version (2004) at http://www.cs.pdx.edu/~singh/ftp/sigcomm03.pdf.
24. Gupta M, Grover S, Singh S. A feasibility study for power management in LAN switches. *Proceedings of the 12th IEEE International Conference on Network Protocols*, October 2004. ■