Greener PCs for the Enterprise

Bruce Nordman, *Lawrence Berkeley National Laboratory*
Ken Christensen, *University of South Florida*

PC energy use is a growing cost to enterprises, with most machines remaining fully powered on, even while idle, for most of the time. The Network Connectivity Proxy maintains network presence for PCs but lets them sleep while idle, thus saving energy and reducing total cost of ownership.

One of the most urgent challenges of the 21st century is to engineer new technologies that can transition us to a more sustainable society with a reduced CO_2 footprint. PC system units and monitors comprise 3 percent of the total electricity consumed in the US, or roughly 100 TWh/year. At a rate of $0.10 per kWh, the annual cost of this electricity is $10 billion. We all know that electricity costs are a major factor in the cost of operating a data center, but electricity costs are also significant for enterprise PCs. Enterprise, or office, PC system units and monitors consume approximately 65 TWh/year, which is roughly 5 percent of commercial building electricity; residential systems consume just over 30 TWh/year.[1] Figure 1 shows PC energy use (including monitors) in the larger context.[1,2] PCs—both in the enterprise and at home—often remain fully powered on, even when inactive or idle. It's during these idle periods that we can achieve the greatest energy savings.

Being connected to the Internet requires some form of active participation—when hosts fail to respond, they "fall off the network," and applications fail. Consequently, network hosts consume billions of dollars worth of electricity every year to stay fully powered on at all times for the sole purpose of maintaining network connectivity or "presence." In fact, researchers have found that roughly 60 percent of office desktop PCs are left on continuously.[3] If not for the need for network connectivity, most of these hosts could be asleep most of the time, with significant energy savings resulting. This need to maintain network connectivity also contributes to why people disable existing power management features in many PCs. We can remove this barrier to PCs using an energy-saving sleep mode when they're idle in one of two ways: by redesigning network protocols and applications or by encapsulating the intelligence for maintaining network presence in an entity other than the operating system and

 Published by the IEEE Computer Society 1520-9202/09/$25.00 © 2009 IEEE

applications running in the system CPU. In our work, we've pursued the second option via the Network Connectivity Proxy (NCP), which is an entity that maintains full network presence for a sleeping network host. This article describes how demands for constant network connectivity drive up PCs' powered-on times and how NCP technology can address both the need for network connectivity and the need to reduce energy consumed by PCs.

How PCs Use Energy

Energy is power consumed over time, so the time dimension of electronics energy use is critical. For a PC, *idle power* and *time* (when the machine's on but not performing any user application function) are the most important numbers to know (see the "PC Energy Use" sidebar). Figure 2 shows the power a typical PC consumes. A new PC in sleep mode typically uses just 5 percent of idle power, and one that's turned off a mere 3 percent. Note that the difference between sleep and off is small, but the difference from either to idle is huge. The peak power (when booting up or otherwise doing as much work as possible) can be 50 percent above the idle level, but this rarely occurs. Because most PC usage requires just a small part of the system's total compute capacity, the average on power is usually close to the idle level.

Two major changes can reduce overall PC energy use (and that of electronics generally):

- PC manufacturers can use more energy-efficient components—in particular, more energy-efficient processors. This is most important for idle because it covers so much time, and the power levels are so much higher than sleep or off.
- Users can change their usage patterns. It's well worth investing in small increases in the sleep power level to leverage large shifts of time from idle to sleep.

In addition, the mix of devices can be changed as long as doing so doesn't compromise the service needed. In some computing environments, for example, thin clients (plus a server) or notebooks can replace desktop PCs. However, thin clients usually save the most costs in terms of equipment purchase and management, with modest energy savings; switching to notebooks saves energy but usually does not reduce the total cost of ownership (TCO). Nevertheless, changing both power levels and usage patterns will reduce PC energy use—the big question is the balance between them. Power-level reductions have dominated past efforts, but the most potential savings (at least those that are cost-effective on energy savings alone) come from changing usage.

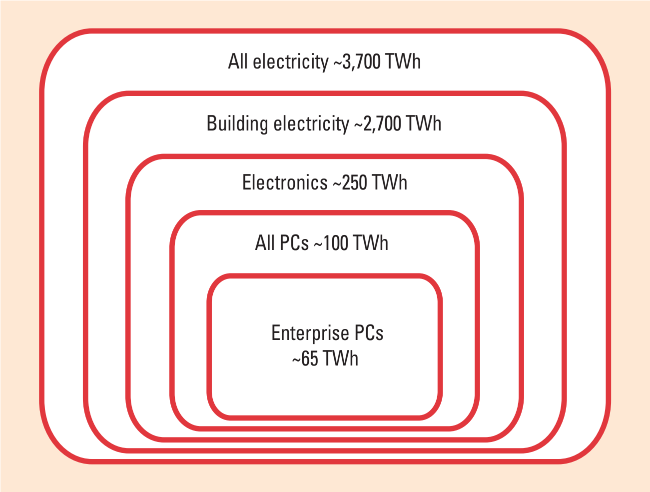

Figure 1. US energy usage in 2006. Although this illustration isn't to scale, PC energy use (including monitors) represents a sizeable chunk of the overall total.

The usage in Figure 2 is an average of many desktop PCs deployed in different ways, with the key variable being their power state over night and during weekends (three-fourths of the total hours in a week). Many desktop PCs remain on continuously, which leads to the large annual idle time shown in the figure. Systems that do sleep spend substantial time in that state, but most don't sleep at all, with some routinely powered down to off. Many systems are left on over night and on weekends, thus most idle time occurs when no one is present—ideal times for the PC to be asleep. With this usage pattern, universal use of sleep would cut PC energy use by more than 50 percent.

User confusion is one reason why PCs are left on during times of nonuse. Some people believe that "screensavers" save energy, but, in fact, the reverse is true: they prevent display power management and keep the CPU more active than it would be with a static desktop screen. Other users observe the display going to sleep and assume that the whole system is powered down, even if the PC itself is still fully up. Finally, some people don't understand the device's power modes or how to assess them visually (see "The Power Control User Interface" sidebar).

PC Energy Use

It's important to understand how much energy a PC uses and the possible impact of this energy use.

What Is Power and Energy?

Power is a measure of instantaneous work, with electrical power measured in watts (W). Energy is power over time, measured in watt-hours (Wh). One thousand Wh, or 1 kWh, is the most convenient measure of electricity consumer use. The average cost of 1 kWh in the US is roughly $0.10; the average US residence uses 10,700 kWh/year.

How Much Energy Does a Single Desktop PC Use?

A typical existing desktop PC system unit uses roughly 80 W when idle, and a typical 17-inch LCD display uses roughly 35 W when powered on.[1] Thus, the cost to operate a single 115-W PC and monitor fully powered on 24 hours a day, seven days a week, for one year is roughly US$100 (this is 8,760 hours/year x 115 W x $0.10/kWh). Active energy (above the idle level) contributes just a few percent to a PC's annual total.

What Is the Climate Impact of PCs?

When electricity is generated from fossil fuels, CO_2—a greenhouse gas that contributes to global warming—is released into the atmosphere. In the US, 70 percent of electricity comes from fossil fuels.[2] The average amount of CO_2 released per kWh is 0.7 kg; the total CO_2 entering the atmosphere attributable to all PCs and monitors in the US is roughly 70 million metric tons per year. The US Environmental Protection Agency's greenhouse gas equivalencies calculator (www.epa.gov/cleanenergy/energy-resources/calculator.html) computes other equivalences.

How Much Energy Used by PCs Could Be Saved?

Researchers have estimated that more than half of the energy used by PCs occurs when no one is present. Figure 2 in the main text shows that a typical business PC is idle most of the time.[3] New desktop PCs have lower power levels in all modes than the typical existing PC—roughly 60 W while on, 2.5 W in sleep, and 1.5 W while off.[1] The energy used during this time could be saved if the PC were instead asleep, with total energy savings for a single PC per year equaling 400 kWh, or $40. For a company with 10,000 such PCs, the yearly savings would be roughly $400,000.

What about non-PC Equipment?

Other IT and consumer electronic devices as well as set-top boxes, game consoles, printers, copiers, and so on are energy consumers, too. Lawrence Berkeley National Laboratory has studied energy usage of office equipment for 20 years and estimates that office PCs and monitors consume roughly 65 TWh per year, with those in residences adding approximately 30 TWh per year more.[1]

References

1. M.C. Sanchez et al., "Savings Estimates for the United States Environmental Protection Agency's ENERGY STAR Voluntary Product Labeling Program," *Energy Policy*, vol. 36 no. 6, 2008, pp. 2098–2108.
2. *Energy Information Administration*, "Electric Power Monthly," US Dept. Energy, 2009; www.eia.doe.gov/cneaf/electricity/epm/table1_1.html.
3. J. Roberson et al., *After-Hours Power Status of Office Equipment and Inventory of Miscellaneous Plug-Load Equipment*, tech. report LBNL-53729-Revised, Lawrence Berkeley Nat'l Lab., 2004.

Another reason for systems to be left on is long boot times (and the loss of state when shutting down the operating system), coupled with an ingrained aversion to using sleep. Perhaps the user had a bad experience with a system that went to sleep but didn't reliably wake up. Or, maybe this person doesn't know how to manually engage sleep or set the system to automatically go to sleep. When the choice is between off (with a slow boot) and leaving a machine on continuously, some users will choose the off option, but most will simply leave the machine on (see the "Off Is *So* Last Century" sidebar).

Finally, simple inertia is another major reason for PCs being left on—that is, people might have started leaving their PCs on all the time and continued to do so without reexamination, even as the reasons for the initial habit have long passed.

The Need for Network Connectivity

Specific applications and usage models drive the need for persistent network connectivity, but they rely on an infrastructure of basic network protocols to work. A simple example of a needed basic function for connectivity is the Address Resolution Protocol (ARP), which associates an Ethernet or Wi-Fi hardware (or MAC) address with the IP address of each PC and router port on a subnet. Each PC (and router port) maintains an ARP cache that contains the IP and MAC addresses for every PC on the subnet. In the router, the ARP cache refreshes by periodically broadcasting ARP requests to all hosts on the subnet, but if a PC is sleeping, it can't reply to such requests. This missing reply means that the entry for the sleeping PC in the router table will be timed-out and removed, which has an extremely serious consequence: the router can no longer forward packets from outside the subnet to the sleeping PC. Quite simply, the router no longer knows how to address packets to the nonresponding host. The router can't even deliver a special wake-up packet, so the sleeping PC remains effectively unreachable to the rest of the world (outside its subnet).

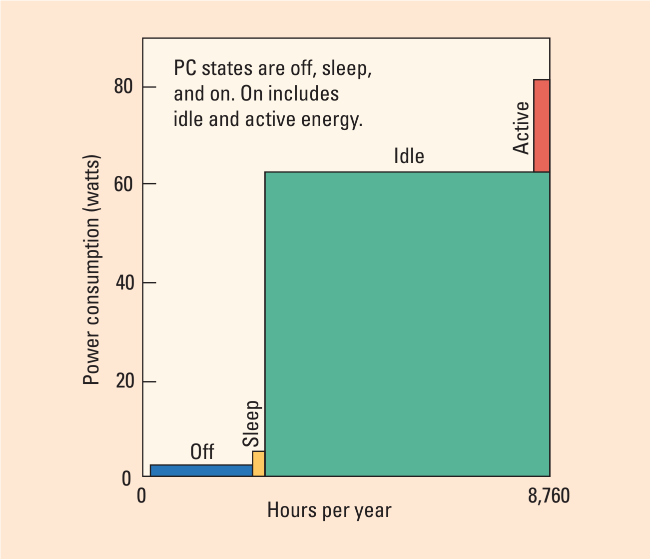

Figure 2. Enterprise PC usage patterns. A typical office computer consumes most of its energy when it's doing nothing useful, such as when idle.

The Power Control User Interface

Consistent user interface elements are common in daily life, from telephone keypads to automobile controls to media player play/pause/rewind controls. These interface elements make products easier (and safer) to use and typically cost no more than inconsistent controls would.

In the past, many electronic devices used inconsistent interface elements in power controls, resulting in needless confusion about what power state a device was in or how power management could be configured.

IEEE 1621 specifies user interface elements such as symbols, indicator lights, and terms for controlling the power status of electronics. These elements appear on the outside of the product (typically, near switches), and in software control panels and documentation. Indicator lights have color coding (green for on, yellow for sleep, and nothing for off). This standard clarifies that the basic power modes for electronics are on, sleep, and off, so that we can look forward to a time when the inherently confusing word "standby" no longer appears in any user interface.

For IT professionals, this makes communication with users easier, but inevitably involves some reeducation and changing some user beliefs and expectations.

Off Is *So* Last Century

Many people almost never turn off their PCs; typically, PCs might be asleep most of the time, but only fully off for unusual travel or maintenance reasons. Moreover, modern PCs can wake quickly (for example, Microsoft Vista specifies less than two seconds out of the box), making the user inconvenience (or "annoyance cost") of using sleep mode low.

Sleep does require more power than off, but since the difference is small (less than 2 watts for most recent PCs), the energy cost to shift from off to sleep mode is small. Most sleeping systems use less total energy than those routinely powered off. This seems counterintuitive, but the reduced time in idle from frequent use of sleep will account for much more savings than the energy required to be asleep rather than off.

This principle also extends to monitors, as many use virtually the same power in sleep mode or off, so the effort used to manually turn them on and off would be better expended elsewhere. This is important to IT managers as they'll need to educate their users to shift from off to sleep as the primary low-power mode.

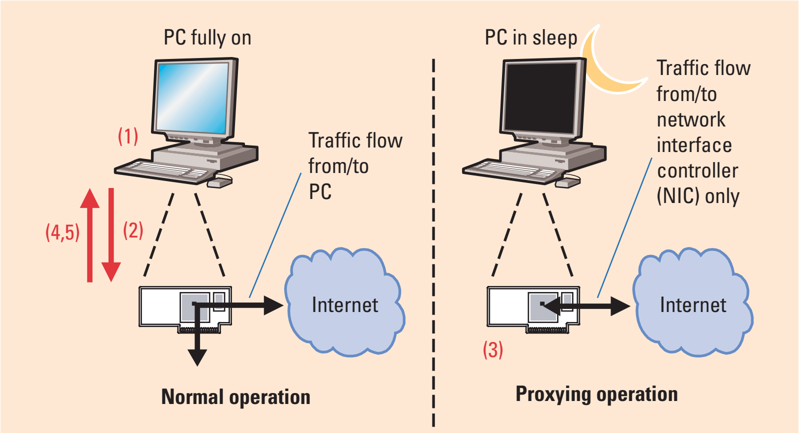

Figure 3. The SmartNIC with proxy capability. Red arrows show transfer of state information between the PC's operating system and the proxy.

Beyond ARP, many other protocols must also be considered. To maintain network connectivity, a PC must be able to support several application and protocol primitives, including the following:

- For IPv4, maintain host-level reachability by responding to periodic ARP requests; for IPv6, maintain reachability by responding to neighbor solicitation (NS) messages.
- Maintain its IP address by generating periodic Dynamic Host Control Protocol (DHCP) lease requests.
- Maintain its manageability by responding to Internet Control Message Protocol (ICMP) packets, such as ping.
- Support NetBIOS name resolution by responding to NetBIOS name queries as appropriate (if running NetBIOS protocol and applications).
- Maintain application-level reachability by responding to TCP SYN packets sent to open (listening) ports.
- Maintain or preserve application state (for example, current user workspace and data) for any applications with open long-term TCP connections.
- Maintain or preserve application state by responding to any number of application-level messages, including heartbeat messages and specific requests for service.

The last category might include support for instant messaging (IM) heartbeats, virtual private network (VPN) tunnels, network address translation (NAT) mappings, universal plug and play (UPnP) discovery, Session Initiation Protocol (SIP) requests, Simple Network Management Protocol (SNMP) requests, Internet Group Management Protocol (IGMP) requests, IP security (IPsec) key maintenance, peer-to-peer (P2P) queries, and many other application and higher-layer protocols. A broad range of future applications under the rubric of rich Internet applications (RIAs) might also have connectivity requirements. RIAs are Web-based applications in which the Web client executes the user interface, and the back-end application server executes the application itself. RIAs support both pull and push of data via TCP connections and potentially split state between a server and client. RIAs could have a major impact on power management in desktop PCs; research is still under way to fully define connectivity requirements for enterprise PCs. PCs on Wi-Fi networks must be able to maintain association with an access point (AP) by using keep-alive mechanisms and executing key management protocols, as appropriate.

The Network Connectivity Proxy

The NCP is an entity that implements a network host's key presence capabilities to let the PC sleep yet appear to other devices to be fully operational and connected. PCs connected to the network maintain their presence to other systems by correctly generating and responding to messages from both network protocols and applications. In other words, the PC sleeps when it can and wakes up only when it needs to.

Powered-down devices require reliable and standard wake-up to return to a fully powered state when needed. However, most network messages destined for a system don't require a device's full resources, such as a desktop PC's powerful (and power-hungry) processor and significant amounts of memory and storage. The NCP can't cover all protocols and applications, but saving energy on most systems isn't undermined by a small percentage of PCs remaining on 24 hours a day, seven days a week. However, the NCP could potentially support energy-efficient operation of many RIAs by maintaining connections and possibly buffering data to let the client host sleep.

How Proxying Works

A proxy performs four basic functions: responding to routine requests, automatically generating routine protocol messages, identifying when a wake up is truly warranted, and ignoring all other packets. A network connectivity proxy's operation requires cooperation with the operating system and data from a few key applications. When the PC is awake, the proxy does nothing: it only operates when the PC is asleep (a PC can enter sleep mode based on a predefined period of user inactivity or by manual intervention, such as by closing a notebook's lid). The NCP covers for a sleeping host, so it needs to know when the host's power state changes (going to sleep or waking) and be able to transfer state between the host and the proxy.

Accordingly, the NCP's key functional steps look like this (see Figure 3):

1. The operating system determines that it's time to go to sleep.
2. The operating system passes state to the proxy; the PC goes to sleep.
3. The proxy maintains "full network connectivity" (generating protocol and application packets as needed).
4. The proxy determines when a packet requiring wake up has arrived; it then signals the PC to wake up.
5. The proxy waits for the PC to fully wake up; the proxy passes state back to the operating system, and then the PC returns to normal network operation.

In addition, the PC might set a real-time-clock-wake event for periodic events that it can't delegate to the proxy, or the system might wake up based on user activity.

When a device wakes up, it often experiences a delay from when the wake-up signal occurs (whether internally generated or from the network) and when the system is fully ready to receive and respond to network queries. For older PCs, this can be on the order of 10 seconds or more—more modern ones can take just a few seconds (in fact, Microsoft now specifies that Windows PCs running Vista will wake up in less than two seconds). The proxy can buffer the wake-up and successive packets received and forward them once the PC is awake. Network protocols generally have mechanisms to retry sending packets when they aren't acknowledged because applications and higher-layer protocols on IP networks assume some degree of unreliability. Thus, accommodating the waking time is feasible for current applications and protocols.

Proxy Location

The simplest place to put the proxy is within the network interface controller (NIC), which greatly reduces configuration problems, eases the passing of state information, and might be essential for mobile devices. However, an alternative is for the proxy to reside in the network device immediately adjacent to the sleeping PC. This is most commonly a wiring closet first-level switch or router or a Wi-Fi AP. The PC's operating system would have to be proxy-capable, but existing PCs could be proxied this way—the only hardware requirement would be the existence of a wake-on-LAN (WOL) capability—that is, the capability to wake up a sleeping PC via a special network request. An additional requirement would be that the operating system and proxy subsystem both implement the same protocol for passing state back and forth.

A proxy performs four basic functions: responding to routine requests, automatically generating routine protocol messages, identifying when a wake up is truly warranted, and ignoring all other packets.

Wake-on-LAN

Most existing methods for PCs to wake on selected network traffic fall under the general WOL description (see the "Reducing Energy Use with Power Management" sidebar). As the list of protocols described earlier makes clear, proxying involves a considerable amount of responding to and sending of packets rather than simply generating system wake ups, which is very different from WOL.

Some organizations successfully use WOL to bring machines to a fully on state prior to being

Reducing Energy Use with Power Management

Researchers have addressed the contribution of energy use to the total cost of PC ownership in the enterprise since the mid 1990s to varying degrees. So far, they've identified three approaches to reducing energy used by existing PCs through better use of power management.

Support for Remote Wake Up

The need to be able to wake up sleeping PCs remotely via a network interface has long been recognized and generally referred to as wake-on-LAN (WOL). In the 1990s, AMD and several other companies developed the magic packet technology to do just this. A magic packet-enabled PC has an Ethernet network interface controller (NIC) that's always powered on, even when the rest of the PC is in a sleep state. The NIC is programmed to recognize a specially defined magic packet, and when it receives such a packet, trigger an interrupt to wake up the PC. Magic packet has been extended to directed packet match, in which specific protocol packets can trigger a wake up.

Global Control of Enterprise PC States

Several companies have recognized the business opportunity of saving energy, and thus money, in enterprises by controlling PC wake and sleep times, including Verdiem, 1e, and BigFix, all of which market products that enable a single manager to globally control the power management settings of all the desktop PCs in a single enterprise (Energy Star provides a simpler version of this capability for free). Although global control is feasible in some settings, it isn't as flexible or far-reaching as would be a fully distributed method of putting PCs not in use to sleep.

Programs for Tuning Power Management Settings

Desktop PC operating systems, such as Microsoft Windows, let users configure their own power management settings, including the amount of idle time before a PC goes to sleep. The default is often 30 minutes, which means the PC remains idle for 30 minutes before going to sleep. Decreasing this time can increase possible energy savings, and several programs and tools can now let users configure these settings as a function of the time of day or otherwise enable and disable power management without having to dig into arcane configuration menus. Two such programs are Verdiem's Edison application (www.verdiem.com/edison/) and Google's Energy Saver gadget (http://desktop.google.com/plugins/i/energysaver.html).

needed, but this is cumbersome in several respects. For manual updates, for example, it adds an extra step to every maintenance process; for automatic updates, sending a WOL packet must become part of the overall process and include a suitable delay for the system to wake up. Another issue is that WOL packets aren't routable, so the sending and receiving systems must be on the same subnet. Ways around this problem exist, but they have their own roadblocks—for example, it's possible to run a special application on a single PC in each subnet that can receive WOL requests for that subnet and then relay a WOL packet to the target system. Or the user or network administrator can configure the system to wake up on a directed packet match rather than on just the magic packet, a specially defined packet intended to wake up sleeping PCs via a network interface. However, in most computing environments, directed packet wake-up patterns do not allow energy-savings settings that wake systems up enough to maintain sufficient functionality tend to make them wake up too often to gain sufficient sleep time. Similarly, those with settings that enable worthwhile sleep sacrifice too much functionality.

It's widely recognized that WOL alone isn't sufficient for achieving significant energy savings in enterprise PCs. Some time ago, the Distributed Management Task Force (DMTF) Alert Standard Format (ASF) standard specified a facility for a NIC to respond to ARP requests. More recently, Microsoft has described features planned for its Windows 7 release that offload some response to network traffic from the operating system to the NIC—notably, enhanced pattern matching for wake up and support for ARP in NICs. Intel recently introduced a "remote wake" capability in its motherboards for SIP softphones that enables PCs running a SIP softphone to maintain connectivity with a SIP server and wake upon receiving an incoming SIP call request.

The NCP and Energy Star

The US Environmental Protection Agency (EPA) approved the newest version of the Energy Star specification for computers in November 2008, which went into effect in July 2009. The "Energy Star Program Requirements for Computers, Version 5.0" specifically includes a provision for PCs that support proxying (www.energystar.gov/index.cfm?c=revisions.computer_spec):

> Full Network Connectivity: The ability of the computer to maintain network presence while in sleep and intelligently wake when further processing is required (including occasional processing required to maintain network presence). … From the vantage point of the network, a sleeping computer with full network connectivity is functionally equivalent to an idle computer with respect to common applications and usage models. Full network connectivity in sleep is not limited to a specific set of protocols but can cover applications installed after initial installation.

Proxying PCs are assumed to spend considerably less time in idle than their conventional counterparts, so the evaluation criteria allow more flexibility in idle power. The EPA doesn't define the specifics of full network connectivity, but

> [f]or a system to qualify under the proxying weightings above, it must meet a non-proprietary proxying standard that has been approved by the EPA and the European Union as meeting the goals of ENERGY STAR.

Ecma International—most famous for standardizing JavaScript—is hosting the effort to implement a standardized version 5.0 Energy Star proxying definition. The Ecma TC32-TG21 committee focuses on "Proxying Support for Sleep Modes," and has the participation of leading hardware, operating system, and PC OEM companies (www.ecma-international.org/memento/TC32-TG21.htm). A standard is expected in late 2009, with the expectation that the Energy Star program will then designate it as meeting the program's definition of "full network connectivity" and so enable it to be used to qualify PCs under the Energy Star label. For further information on proxying and related topics, see http://efficientnetworks.lbl.gov.

Other Work Related to Proxying

With our colleagues, we first explored proxying to maintain networking connectivity for sleeping hosts in the late 1990s, specifically for ARP in shared Ethernet networks.[5] We refined this work in the early 2000s and have further developed it in the past two years.[6–8] These and similar reports on proxying also appear with the Energy Efficient Internet Project (at the University of South Florida and University of Florida, funded in part by the US National Science Foundation and Cisco Systems), which studies ways to improve the Internet's energy efficiency by focusing on the most basic yet often neglected energy consumers—edge devices. See www.csee.usf.edu/~christen/energy/main.html for more information.

Ecma International—most famous for standardizing JavaScript—is hosting the effort to implement a standardized version 5.0 Energy Star proxying definition.

Researchers recently presented an initial exploration of the architectural constructs required to support selective connectivity at ACM HotNets 2007. Selective connectivity is the notion that a host can choose the degree to which it maintains a network presence, rather than today's binary "connected" or "disconnected" modes. A key architectural construct to support selective connectivity is an assistant that stands in for a host that's asleep.

In industry, proxying to enable power management already exists for at least one specific protocol—UPnP, which uses a fully distributed discovery protocol that requires all devices in a UPnP network to be fully powered up at all times to respond to discovery messages. In August 2007, the UPnP Forum released its UPnP low-power architecture, version 1.0 (www.upnp.org/specs/lp.asp). To sleep and still be discoverable by UPnP control points, the architecture defines a power management proxy specific to UPnP only. However, proxying for UPnP isn't transparent: it requires changes to UPnP client functionality. This solution for enabling power management in one specific protocol highlights the need for a more general—and transparent—approach. The DMTF Alert

Standard Format 2.0 specification describes proxying of ARP in Ethernet NICs, and many NICs currently installed in PCs already support it. In 2008, Microsoft Research demonstrated a prototype proxy called Somniloquy, whereby a secondary low-power processor covers for the PC's main processor. Researchers developed the prototype on a USB-based gumstix device; so far, Somniloquy appears to be a general-purpose architecture that can also support network applications—including BitTorrent downloads—via application stubs.

Proxying will probably save the most PC energy use in coming years, but other efforts will also help, including Energy Efficient Ethernet. EEE relies on the fact that most Ethernet links have very low utilization most of the time (in terms of actual data transmitted as a percentage of link capacity). When both ends of a link are EEE-capable, the physical-layer power is greatly reduced under normal operating conditions, saving a watt or more of power for 1 Gb/s links on PCs (and many times that for those running at 10 Gb/s in data centers and network equipment). The IEEE is standardizing this through its 802.3az task force (http://ieee802.org/3/az), with late 2010 as the expected time frame for the standard to receive all approvals. Unfortunately, EEE-capable PCs won't achieve full energy savings until the legacy wiring closet switches to which they're connected are replaced or upgraded.

Practical Impacts to the IT Manager

PCs with the hardware and software infrastructure for proxying (and the software for implementing it on network equipment) could become available in 2010. In the meantime, it's important for IT professionals to identify any usage models or applications that proxying can't support, so that users and systems that rely on them are treated separately from the bulk of people who can use proxying successfully for their computing needs. IT managers will need to ensure that standard disk images have power management enabled and come up with a way to describe proxying to ordinary users so that they understand how to use sleep mode correctly.

As we've mentioned, proxying attempts to hide the fact that the PC is asleep from the rest of the network, which is the best course of action most of the time. However, select applications might want to disclose the PC's power state to specific parts of the network to ensure that they can make the best resource allocation decisions. One of us (Christensen, with his student Francisco Blanquicet) has proposed a power MIB for SNMP to disclose power state and other energy-related information. Power-aware applications can use this information to make decisions that respect users' desire to save energy yet take into account system latency for full wake up. P2P and other emerging enterprise applications particularly need to be power-aware because they can have the unintended effect of requiring systems to stay awake when for all other purposes they don't need to be.

A key design goal of proxying is that it be invisible to applications and users. As such, we don't believe that its deployment will require much intervention from IT managers or staff. It's doubtful that 100 percent of all future PCs will operate with proxying enabled; there will always be exceptions, such as those PCs used in server roles (in which frequent, constant access is expected, and response time performance is a critical measure). We envision proxying as being applicable and useful to the vast majority of enterprise (and home) PCs, but not to critical servers in a data center. Once the proxying standard is in place, then new protocols and applications can be designed so as to not "break" when the proxy engages, and thus be compatible with PCs routinely going through sleep cycles at night and during the work day.

For a typical new desktop PC, the power savings from shifting from idle to sleep is just under 60 W. For a PC that stays on 24 hours a day, seven days a week without proxying—but that can sleep for three-fourths with it—annual savings amount to roughly 400 kWh, or US$40 at $0.10/kWh (per PC).

Although the most immediate energy savings will be in desktops, proxying technology applies equally to notebooks and printers. While printers can and do go to sleep (principally by powering down their fuser unit and imaging electronics), the processor that handles network connectivity must stay fully on. Nevertheless, proxying could still notably reduce the power levels required in sleep. In the home context, game consoles and set-top boxes could also make good use of proxying technology. Even devices such as phones, TVs, and other displays could use proxying as they increasingly use Internet connections. Ultimately, we'll need to reduce the energy consumption of all the electronic devices we use, and proxying is a relatively easy and low-cost way to do it. IT

Acknowledgments

This material is based on work supported by the California Energy Commission's Public Interest Energy Research program and the US Department of Energy under contract number DE-AC-02-05CH11231 (Nordman), and the US National Science Foundation under grant number CNS-0520081 (Christensen).

References

1. M.C. Sanchez et al., "Savings Estimates for the United States Environmental Protection Agency's ENERGY STAR Voluntary Product Labeling Program," *Energy Policy*, vol. 36, no. 6, 2008, pp. 2098–2108.
2. Energy Information Administration, "Electric Power Monthly," US Dept. Energy, 2009; www.eia.doe.gov/cneaf/electricity/epm/table5_1.html.
3. J. Roberson et al., *After-Hours Power Status of Office Equipment and Inventory of Miscellaneous Plug-Load Equipment*, tech. report LBNL-53729-Revised, Lawrence Berkeley Nat'l Lab., 2004.
4. Energy Information Administration, "Electric Power Monthly," US Dept. Energy, 2009; www.eia.doe.gov/cneaf/electricity/epm/table1_1.html.
5. K. Christensen and F. Gulledge, "Enabling Power Management for Network-Attached Computers," *Int'l J. Network Management*, vol. 8, no. 2, 1998, pp. 120–130.
6. K. Christensen et al., "The Next Frontier for Communications Networks: Power Management," *Computer Communications*, vol. 27, no. 18, 2004, pp. 1758–1770.
7. B. Nordman and K. Christensen, "Improving the Energy Efficiency of Ethernet-Connected Devices: A Proposal for Proxying," white paper, version 1.0, Ethernet Alliance, Oct. 2007.
8. M. Jimeno, K. Christensen, and B. Nordman, "A Network Connection Proxy to Enable Hosts to Sleep and Save Energy," *Proc. IEEE Int'l Performance Computing and Communications Conf.*, IEEE CS Press, 2008, pp. 101–110.

Bruce Nordman *is a staff research associate in the Energy Analysis Department, Environmental Energy Technologies Division, of Lawrence Berkeley National Laboratory. Contact him at bnordman@lbl.gov.*

Ken Christensen *is a professor in the Department of Computer Science and Engineering at the University of South Florida. Contact him at christen@cse.usf.edu.*