Managing Energy Use in a Network with a New SNMP Power State MIB

Francisco Blanquicet and Ken Christensen
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
{fblanqui, christen}@cse.usf.edu

Abstract— **Energy consumption has become a major factor in the total cost of ownership (TCO) of IT equipment. The power state of IT equipment is effectively "invisible" to the network making it difficult to measure or control energy use. In this short paper we propose, prototype, and evaluate a new SNMP Power State MIB and its agent to expose equipment power state to the network. The power state includes all supported power management (PM) capabilities, current PM settings, total and current active, inactive, and sleep times, and statistics on wake-up and sleep events. With knowledge of the power state of network devices, a network manager could remotely audit the energy consumption of IT equipment and make changes to PM settings. We have implemented a subset of the Power State MIB for Microsoft Windows Vista desktop and server computers.**

Keywords- Network Management, SNMP, Power State MIB

I. INTRODUCTION

The Total Cost of Ownership (TCO) of IT equipment is a sum of hardware, software, personnel, and energy costs. Increasingly, energy costs are becoming a major factor in the TCO of both data centers and enterprise computing. The energy use of IT equipment also has an environmental impact. Organizations such as Climate Savers Computing Initiative are evidence of the great interest in reducing the CO_2 footprint of IT equipment. Desktop computers are a major component of the energy use of IT equipment. The EPA estimates that PCs consume 2% of all electricity in the US [4].

In order to reduce the energy use of IT equipment, there must be a means of monitoring and controlling it. To date, there have been only vendor-specific solutions for monitoring and controlling the amount of time that PCs are on and active, on and idle, sleeping, and off. Thus, the open problem to be addressed is how the power state of IT equipment can be exposed in a standard manner compatible with existing network management protocols and applications. In this paper, we investigate how power state can be exposed – and thus monitored and controlled – using SNMP.

The remainder of this paper is organized as follows. Section 2 is a brief background of power management and SNMP. Section 3 describes the design of our SNMP Power State MIB. Section 4 describes the implementation and evaluation of the new MIB for a Windows PC. Section 5 describes applications, and Section 6 is a summary and describes future work.

This material is based on work supported by the National Science Foundation under Grant No. CNS-0520081.

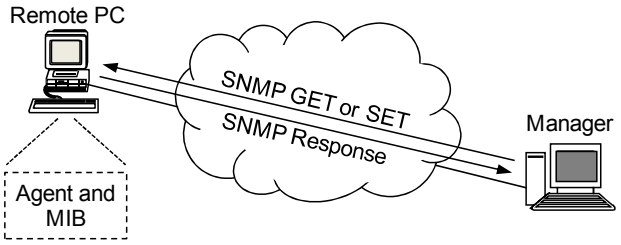

Figure 1. Overview of SNMP showing a Manager and Remote PC

II. BACKGROUND

Most IT equipment, including desktop PCs, maintain internal power state, support power management (PM), and have network management capabilities. However, these three capabilities have not been coupled together.

The Advanced Configuration and Power Interface (ACPI) is an industry standard that defines common interfaces for hardware recognition, device configuration, and power management [1]. ACPI defines seven states for computers: *Working* G0 (S0), *Sleeping* G1 (S1, S2, S3, S4), *Soft-Off* G2 (S5), and *Mechanical-Off* G3. Device states are: *Fully-On* D0, *Intermediate* D1 and D2, and *Off* D3. Processor states are: *Operational* C0, *Halt* C1, *Stop-Clock* C2, and *Sleep* C3. The DMTF defines methods to change the power state of a computer through a standard management interface [7].

Power Management (PM) capabilities include turning off or putting to sleep system components (e.g., hard disk and display) or the entire system after a fixed period of inactivity (lack of user-initiated activity) [6]. The setting of inactivity timer values can have considerable impact on energy savings.

The Simple Network Management Protocol (SNMP) is an application protocol that allows a network manager to GET and SET network management objects in remote devices. Remote devices that support SNMP contain an agent that can access network management statistics and settings and a Management Information Base (MIB) that stores this information. Figure 1 shows the basic operation of an SNMP manager. MIBs are defined for specialized devices and capabilities. Two MIB extensions that are related to power use are the Power Over Ethernet MIB [2] and the Uninterruptable Power Supply MIB [3]. Our work is motivated by and builds upon these two MIBs.

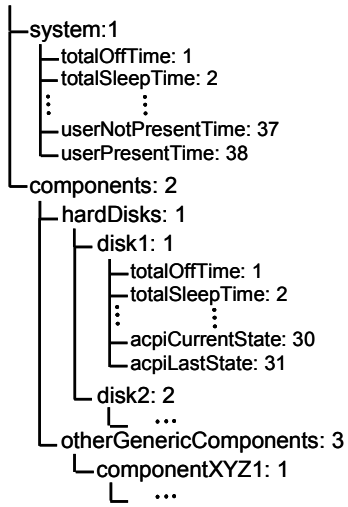

Figure 2. Power State MIB tree description

III. Design of the Power State MIB

The first step in the design of the new Power State MIB was deciding what state objects needed to be exposed for monitoring and what power management parameters needed to be controlled. IT equipment has three standard-defined [5] states, which are:

- On – Power state in which the device has greater (or similar) power consumption, capability, and responsiveness than it does in the Sleep or Off state.
- Sleep – Power consumption, capabilities, and responsiveness are greater than in the Off state, but less than in the On state.
- Off – Power consumption, capabilities, and responsiveness are less than in the On or Sleep states.

The Sleep state can be subdivided into multiple sub-states. When IT equipment is On it can be either active or inactive (idle). When inactive, the equipment could be in a Sleep state to save energy. *A key goal of controlling power management settings is to minimize On-inactive time in favor of Sleep time.* Thus, we need to be able to monitor both total and current On (active and inactive), Sleep, and Off times. We also need to be able to access power management settings (specifically, the inactivity time-out values for system Sleep and Off).

IT equipment typically contains components (e.g., hard disks and displays for a desktop computer) that can be individually power managed. Thus, it is important to monitor components within IT equipment and control their PM settings

TABLE I. Power State MIB objects for the system and its components

NAME	SYNTAX	ACCESS	DESCRIPTOR	UNITS
totalOffTime:1	Integer	Read-Only	Total time in off state	seconds
totalSleepTime:2	Integer	Read-Only	Total time in sleep state	seconds
totalOnTime:3	Integer	Read-Only	Total time in on state	seconds
totalInactiveTime:4	Integer	Read-Only	Total time in inactive state	seconds
totalActiveTime:5	Integer	Read-Only	Total time in active state	seconds
lastOfftime:6	Integer	Read-Only	Duration of last off time period	seconds
lastSleepTime:7	Integer	Read-Only	Duration of last sleep time period	seconds
lastOnTime:8	Integer	Read-Only	Duration of last on time period	seconds
lastInactiveTime:9	Integer	Read-Only	Duration of last inactive time period	seconds
lastActiveTime:10	Integer	Read-Only	Duration of last active time period	seconds
goToSleepCount:11	Counter	Read-Only	Counts number of transitions to sleep state	number
lastGoToSleepType:12	String	Read-Only	Describes what caused last transition to sleep	System/User
goToOffCount:13	Counter	Read-Only	Counts number of transitions to off state	number
lastGoToOffType:14	String	Read-Only	Describes what caused last transition to off	System/User
currentPowerState:15	String	Read-Write	Current state the system is in	Off/Sleep/On
powerSource:16	String	Read-Only	Current power source	AC/Battery
totalTimeACSource:17	Integer	Read-Only	Total time on AC power	seconds
totalTimeBattSource:18	Integer	Read-Only	Total time on battery power	seconds
lastACTime:19	Integer	Read-Only	Duration of last time period on AC power	seconds
lastBattTime:20	Integer	Read-Only	Duration of last time period on battery power	seconds
powerLevel:21	Gauge32	Read-Only	Current power consumption level	Watts
energyConsumption:22	Gauge32	Read-Only	Total energy used	Watt-hours
energyLeft:23	Gauge32	Read-Only	Energy left (if on battery)	Hours
utilization:24	Guage32	Read-Only	Current utilization	percentage
inactivityOffTimer:25	Integer	Read-Write	Inactivity time-out to go to Off state	seconds
inactivitySleepTimer:26	Integer	Read-Write	Inactivity time-out to go to Sleep state	seconds
currentInactivityValue:27	Integer	Read-Only	Current inactivity timer value	seconds
acpiSupportedStates:28	String	Read-Only	ACPI supported states	ACPI defined states
acpiEnabledStates:29	String	Read-Only	ACPI enabled states	ACPI defined states
acpiCurrentState:30	String	Read-Only	ACPI current state	ACPI defined states
acpiLastState:31	String	Read-Only	ACPI last state	ACPI defined states

TABLE II. Power State MIB objects that apply to the system only

NAME	SYNTAX	ACCESS	DESCRIPTOR	UNITS
wolSupportedTypes:32	String	Read-Only	WOL supported types	Magic/Directed Packet
wolEnabledType:33	String	Read-Write	WOL enabled type	Magic/Directed Packet
wakeUpCount:34	Counter	Read-Only	Counts times the system woke-up b/c WOL	number
lastWakeUpType:35	String	Read-Only	Describes last type of WOL packet received	Magic/Directed Packet
userPresence:36	Boolean	Read-Only	Determination if user is present	true/false
userNotPresentTime:37	Integer	Read-Only	Time the user has been gone	seconds
userPresentTime:38	Integer	Read-Only	Time the user has been present	seconds

as well. Components within desktop PCs, and some other IT equipment types, support ACPI states. These states – which are supported and enabled, and what are the current and last states – should be able to be monitored. The new Power State MIB needs to be general enough to support new components not yet in existence.

The new power state MIB has two groups and 38 objects. Figure 2 shows these groups, and Tables I and II describe the MIB objects. The 38 objects encapsulated by the Power State MIB can be divided as follows: system-only objects and system-and-component objects. Here, a device is understood to be a system plus components. Figure 2 shows a possible configuration for the objects inside a device (such as a PC) in a tree structure. Objects 1 through 38 make up the system objects, objects 1 through 31 only apply to components within the device. Table I and Table II provide a complete description of all the objects in the new Power State MIB.

Objects 1 through 14 provide information about the pattern of use of a device. Objects 15 through 24 have information related to the type of power source operating the device. Objects 25 through 27 present PM information similar to the one found in the PM settings of a desktop PC. Objects 28 through 31 expose ACPI information. Objects 32 through 35 provide information about the WOL capabilities of the device, and objects 36 through 38 provide information about the users.

IV. Implementation and Evaluation

An SNMP extension agent was designed for the Windows operating using Microsoft's SNMP API. This agent provides information for the following objects: totalSleepTime, totalInactiveTime, totalActiveTime, lastInactiveTime, and lastActiveTime. The agent is comprised of two sections and is an extension agent to the native Windows SNMP service. The first section is based on an idle tracking process which tracks system activity (i.e., keyboard or mouse activity) every minute. The second section of the agent stores and retrieves the collected information to and from the new SNMP Power State MIB. Our extension agent was installed on a Windows Vista desktop PC.

Using our new Power MIB, we were able to determine what computers were left on at night and see that they were idle. We could also infer that the PM settings were disabled.

V. Application of Power State MIB

We envision two key applications of the Power State MIB. The first is for energy use audits of IT equipment installed in large enterprises. Such audits would discover what equipment exists in the enterprise, its general activity patterns, the level of enablement of power management, and even the precise energy used (if the energyConsumption object is implemented with an internal power meter). The second application is real-time control of power management settings in response to measured activity of IT equipment. Thresholds could be established at which point an SNMP TRAP message could be sent to alert a manager of energy waste (e.g., for IT equipment that is idle, but has been in the On state for a long period of time).

VI. Summary and Future Work

A new Power State SNMP MIB for exposing the power state of network-connected devices has been developed and partially implemented. Exposing power state is the first step towards being able to monitor and control – that is, fully manage – energy consumption of IT equipment. Managing energy consumption is becoming increasingly critical to reducing TCO of IT equipment in data centers and enterprises.

Future work consists of completing the implementation of the Power State MIB for Windows Vista. We also want to explore how SNMP access can be maintained for sleeping devices – for example, by an outboard SNMP proxy. Finally, we want to explore directions towards standardizing a Power State MIB, or incorporate ideas for power state monitoring and control in existing MIBs or other standard network management protocols as they evolve in the future.

Acknowledgement

The authors would like to thank Bruce Nordman from Lawrence Berkeley National Laboratory for his help.

References

[1] Advanced Configuration and Power Interface Specification, Revision 3.0b, October 10, 2006. URL: http://www.acpi.info/spec.htm.

[2] A. Berger and D. Romascanu, "Power Ethernet MIB", RFC 3621, December 2003.

[3] J. Case, "UPS Management Information Base," RFC 1628, May 1994.

[4] "EPA Announces New Computer Efficiency Requirements," EPA Newsroom, Release date: October 23, 2006, Contact: Enesta Jones.

[5] IEEE Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments, IEEE Std 1621-2004, June 8, 2005.

[6] Optimizing Windows Vista Platforms for Energy Efficiency. Microsoft Corporation, May 1, 2007.

[7] Power State Management Profile, DMTF, Document Number: DSP1027, Version 1.0.0.c, February 13, 2007.