

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Sistema de Controle de Consumo para Redes de Computadores

por

ROGER AL-ALAM KROLOW

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de Mestre
em Ciência da Computação

Prof. Ingrid Eleonora Schreiber Jansch-Pôrto
Orientadora

Porto Alegre, outubro de 2000.

CIP - Catalogação na Publicação

Krolow, Roger Al-Alam

Sistema de Controle de Consumo para Redes de Computadores / por Roger Al-Alam Krolow. - Porto Alegre: PPGC da UFRGS, 2000.
109 f.:il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR-RS, 2000. Orientadora: Jansch-Pôrto, Ingrid Eleonora Schreiber.

1. Controle de Consumo. 2. SNMP. 3. UPS. 4. MIB. 5. Gerência de redes.
6. Tolerância a falhas. I. Jansch-Pôrto, Ingrid Eleonora Schreiber.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Henneman

Superintendente de Pós-Graduação: Prof. Philippe Olivier Alexandre Navaux

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenadora do PPGC: Profa Carla Maria Dal Sasso Freitas

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Diversas pessoas foram importantes para a realização deste trabalho, algumas direta e outras indiretamente. É hora de mencionar as pessoas que participaram da minha vida nos últimos anos.

Primeiramente gostaria de agradecer a meus pais, Arlindo e Gilda, e a meus irmãos, Everton e Adriane, pelo apoio e incentivo dados durante todo esse tempo. Também não achei fácil ficar longe esse tempo todo. Valeu pessoal!

Agradeço à Ingrid a oportunidade de realizar um trabalho prático. Agradeço seu empenho para que existissem condições propícias para o desenvolvimento do trabalho - sua preocupação com bolsas, auxílios e equipamentos. Sua orientação exigente foi determinante para o sucesso do trabalho. Obrigado professora.

Peres, Torgan e Clairton, companheiros do laboratório 122, valeu a força, as conversas, as contribuições, os mates. Será que aquela cervejinha ainda sai?

Cechin, te julgava bastante teórico. Grato engano. Tua contribuição foi decisiva no momento de definir o trabalho.

Professor João Netto, obrigado pelos conselhos importantes e pelo acesso à cópia do HP Open View que seu grupo dispunha.

Aos colegas, funcionários e amigos da UFRGS. Pessoas que tornam o Instituto um lugar aprazível. A Eliane na portaria, a Kika no comando da rede, o pessoal da biblioteca, Luís Otávio sempre prestativo, seu Astrogildo na ronda à noite e nos finais de semana. Por favor, organizem mais festas de integração como a que ocorreu em Novembro de 99, no aniversário do Instituto. Grande momento para se conhecer melhor pessoas com quem mantemos contato diário.

Colegas com quem tive mais contato, os “estrangeiros” Jatene, Márcio, Otávio, Roni e Sérgio; os gaúchos Cassal, Denis, Jeferson, Juliano, Kratz, Marilton.

À empresa CP Eletrônica S/A pelo importante projeto que mantém com a UFRGS. Estou convicto de que parcerias entre universidade e empresa são benéficas para ambas e para a sociedade.

Ao CNPq pela concessão da bolsa que possibilitou a realização desse trabalho.

Aos integrantes da CaMbAdA, presença diária a qualquer distância (em outras cidades, estados, países ou mesmo na mesa ao lado), compartilhando descobertas, tristezas e alegrias. É um prazer fazer parte desse grupo. Vocês foram uma força muito grande.

Aos integrantes da Muvuca, com quem dividi o teto por um ano. Nunca vi tão pouca cerveja numa geladeira. Que o sucesso profissional compense esse período de privações :-)

Aninha, foi um prazer e um privilégio ter morado contigo. Que saudades do ap! Teus conselhos e o sabor daquele pão ficarão na minha memória para sempre. Valeu amiga!

Julius e Curuca, companheiros de mestrado na UFRGS, e com o Xuxu, de POA no primeiro ano. Conversas, discussões e incertezas. Vencemos gurizada!

Paulo e Eurico, companheiros próximos nessa etapa final. Mais um passo dado para o nosso futuro.

Por fim, agradeço às pessoas que se dedicam a fazer *software* livre ou gratuito, por criarem as ferramentas que foram utilizadas, sempre que possível, no desenvolvimento do trabalho.

Sumário

Lista de Abreviaturas.....	7
Lista de Figuras.....	9
Lista de Tabelas.....	10
Resumo.....	11
<i>Abstract</i>.....	12
1 Introdução.....	13
2 Equipamentos UPS e Softwares Relacionados.....	17
2.1 Classificação das UPS.....	17
2.2 Conexão das UPS à Rede.....	19
2.3 Softwares Comerciais de Controle de Consumo.....	19
2.3.1 Software SiteNet Multilink.....	20
2.3.2 Software PowerChute.....	21
2.3.3 Software SUPSNet.....	23
2.4 Considerações sobre os Softwares Estudados.....	23
2.5 Características do Sistema Proposto.....	24
3 O Protocolo SNMP.....	27
3.1 Gerente.....	28
3.2 Agente.....	28
3.3 Base de Informações de Gerência.....	29
3.4 Protocolo de Gerência.....	29
3.4.1 Operações.....	30
3.4.2 Segurança.....	31
4 Controle de Consumo.....	34
4.1 O Programa Energy Star.....	34
4.2 Mecanismos de Controle de Consumo.....	35
4.3 Computadores Apple Macintosh.....	37
4.4 Estações Sun Microsystems.....	38
4.5 O Padrão Advanced Power Management.....	38
4.5.1 Estados de Energização do Sistema.....	40
4.5.2 Estados de Energização de Dispositivos.....	41
4.5.3 Estados de Energização do Processador.....	41
4.5.4 Transições de Estados de Energização.....	42
4.5.5 Interface de Funções.....	43
4.5.6 Considerações sobre o APM.....	43
4.6 O Projeto OnNow.....	43
4.6.1 Classes de Dispositivos.....	44
4.6.2 Funções de Gerência de Energia.....	45
4.6.3 O Padrão Advanced Configuration and Power Interface.....	46

4.6.4 Estrutura.....	47
4.6.5 Estados de Energização.....	48
4.6.6 Considerações sobre o ACPI.....	50
4.7 Equipamentos sem Suporte à Gerência de Energia.....	50
4.8 Mecanismos dos Sistemas Operacionais.....	51
4.8.1 MSDOS.....	51
4.8.2 Windows.....	51
4.8.3 Linux.....	53
5 Sistema de Gerência Proposto.....	54
5.1 Gerente Principal.....	55
5.2 Gerente Reserva.....	56
5.3 Agentes.....	57
5.4 Sinal de Vida.....	58
5.5 Estados de Consumo das Entidades.....	58
5.6 Comunicação entre Gerente e Agentes.....	58
5.7 Interface Gráfica.....	59
5.8 Modelo de Falhas.....	59
5.8.1 Falhas do Gerente.....	60
5.8.2 Falhas dos Agentes	61
5.9 Segurança.....	61
5.10 Integração com Plataformas de Gerência.....	62
5.11 Base de Informações de Gerência.....	64
6 Descrição do Gerente.....	68
6.1 Composição.....	69
6.2 Funcionamento.....	73
6.3 Configuração.....	77
6.3.1 Arquivo de Configuração.....	77
6.3.2 Arquivo de Comportamento.....	79
7 Implementação.....	86
7.1 Ambientes e Ferramentas Utilizados.....	86
7.2 Experimentos.....	87
8 Conclusões e Perspectivas.....	91
8.1 Conclusões.....	91
8.2 Perspectivas Futuras.....	93
Anexo A Exemplo de Arquivo de Configuração.....	96
Anexo B Exemplo de Arquivo de Comportamento.....	102
Bibliografia.....	106

Lista de Abreviaturas

AC	<i>Altenating Current</i>
ACPI	<i>Advanced Configuration and Power Interface Specification</i>
APC	<i>American Power Conversion</i>
API	<i>Application Program Interface</i>
APM	<i>Advanced Power Management</i>
ATX	<i>Advanced Technology Extension</i>
BIOS	<i>Basic Input Output System</i>
CBC	<i>Cypher Block Chaining</i>
CIM	<i>Compaq Insight Manager</i>
CMIP	<i>Common Management Information Protocol</i>
CMOT	<i>CMIP over TCP/IP</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CPU	<i>Central Processing Unit</i>
DC	<i>Direct Current</i>
DES	<i>Data Encryption Standard</i>
DLL	<i>Dynamic Link Library</i>
DoS	<i>Denial of Service</i>
GNU	<i>Gnu's not Unix</i>
HMAC	<i>Hash Message Authentication Code</i>
HTML	<i>Hyper Text Markup Language</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
IAB	<i>Internet Architecture Board</i>
ICMP	<i>Internet Control Message Protocol</i>
IEEE	<i>The Institute of Electrical and Electronics Engineers, Inc.</i>
IP	<i>Internet Protocol</i>
Java RMI	<i>Java Remote Method Invocation</i>
kW	Kilowatt
kWh	Kilowatt hora
MAC	<i>Media Access Control Address</i>
MD5	<i>Message Digest 5</i>
MIB	<i>Management Information Base</i>
MSDOS	<i>Microsoft Disk Operating System</i>
NMS	<i>Network Management Station</i>

NVS	<i>Non Volatile Sleep</i>
OSI	<i>Open Systems Interconnection</i>
OSPM	<i>Operating System Power Management</i>
PC	<i>Personal Computer</i>
PDU	<i>Protocol Data Unit</i>
RAM	<i>Random Access Memory</i>
RFC	<i>Request for Comments</i>
RMI	<i>Remote Method Invocation</i>
SHA	<i>Secure Hash Algorithm</i>
SMS	<i>System Management Server</i>
SNMP	<i>Simple Network Management Protocol</i>
SNMPv2c	<i>Simple Network Management Protocol version 2 c</i>
SNMPv3	<i>Simple Network Management Protocol version 3</i>
SO	<i>Sistema Operacional</i>
TCP	<i>Transport Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
UPS	<i>Uninterruptible Power Supply</i>
USB	<i>Universal Serial Bus</i>
USM	<i>User Security Model</i>
VACM	<i>Vision Access Control Model</i>
W	<i>Watts</i>

Lista de Figuras

FIGURA 1.1 - Sistema proposto.....	15
FIGURA 2.1 - Classificação das UPS quanto à forma de fornecimento de energia.....	18
FIGURA 2.2 - Conexão da UPS à rede.....	19
FIGURA 2.3 - Tela de configuração de ações do SiteNet Multilink.....	21
FIGURA 2.4 - Tela principal do PowerChute.....	22
FIGURA 2.5 - Tela de configuração de ações do PowerChute.....	22
FIGURA 3.1 - Elementos do modelo de gerência na Internet.....	28
FIGURA 3.2 - Estrutura Básica da MIB.....	29
FIGURA 3.3 - Formato das mensagens do SNMP.....	30
FIGURA 4.1 - Componentes de um sistema APM.....	39
FIGURA 4.2 - Transições de estado em um computador com APM.....	42
FIGURA 4.3 - Estrutura do ACPI.....	47
FIGURA 4.4 - Estados de energização no ACPI.....	48
FIGURA 4.5 - Ligação da chave eletrônica.....	51
FIGURA 5.1 - Sistema Proposto.....	55
FIGURA 5.2 - Estrutura básica do gerente.....	56
FIGURA 5.3 - Composição do agente.....	57
FIGURA 5.4 - MIB para controle de consumo.....	64
FIGURA 5.5 - MIB NetPower.....	66
FIGURA 6.1 - Estrutura do gerente.....	68
FIGURA 6.2 - Relações entre modos de operação, eventos e entidades gerenciadas.....	69
FIGURA 6.3 - Estrutura do gerente.....	71
FIGURA 6.4 - Lista de ações futuras.....	75
FIGURA 6.5 - Descrição de funcionamento do gerente reserva.....	76
FIGURA 6.6 - Sintaxe dos arquivos.....	77
FIGURA 6.7 - Seção de definição de estados.....	78
FIGURA 6.8 - Exemplo de seção states.....	79
FIGURA 6.8 - Rede utilizada como exemplo.....	80
FIGURA 6.9 - Agrupamento de equipamentos.....	81
FIGURA 6.10 - Seção de definição de grupos.....	82
FIGURA 6.11 - Exemplo de seção de definição de grupo.....	82
FIGURA 6.12 - Seção de relações de dependência.....	82
FIGURA 6.13 - Exemplo de seção de relações de dependência.....	82
FIGURA 6.14 - Sintaxe da seção de declaração de eventos.....	83
FIGURA 6.15 - Declaração de eventos.....	83
FIGURA 6.16 - Sintaxe da seção de declaração de modos.....	84
FIGURA 6.17 - Exemplo de seção de declaração de modos.....	84
FIGURA 6.18 - Sintaxe da seção de definição de comportamento.....	84
FIGURA 7.1 - Cenário do experimento.....	88
FIGURA 7.2 - Tela do gerente.....	89
FIGURA 7.3 - Tela do agente.....	90

Lista de Tabelas

TABELA 2.1 - <i>Softwares</i> analisados e seus fabricantes.....	20
TABELA 2.2 - Características dos <i>softwares</i> de gerência.....	25
TABELA 4.1 - Limites de consumo para computadores.....	34
TABELA 4.2 - Limites de consumo para monitores	34
TABELA 4.3 - Limite de consumo integrado	35
TABELA 4.4 - Estados de espera do Windows	52
TABELA 5.1 - Objetos da MIB proposta.....	65
TABELA 5.2 - Estados dos dispositivos.....	67
TABELA 6.1 - Ações definidas.....	74
TABELA 6.2 - Seções do arquivo de configuração	78
TABELA 6.3 - Variáveis da seção Miscellaneous.....	78
TABELA 6.4 - Seções do arquivo de comportamento	79
TABELA 6.5 - Variáveis da seção Equipment.....	80
TABELA 6.6 - Programação de eventos.....	83
TABELA 6.7 - Variáveis da seção Administrator.....	85

Resumo

Este trabalho define e implementa um sistema de controle de consumo para redes de computadores, objetivando aumentar o tempo de operação da rede em caso de operação com recursos limitados e redução de consumo de energia em situações de fornecimento normal.

Na definição do sistema, denominado NetPower, foi estabelecida uma estrutura através da qual um gerente (coordenador) monitora as atividades dos equipamentos vinculados à rede, e determina alterações nos estados de consumo respectivos, de acordo com as necessidades ou atendimento de padrões de otimização. Aos equipamentos podem ser atribuídos diferentes privilégios em uma hierarquia adaptável a diversos ambientes. Um reserva oferece opção às falhas do gerente.

A implementação está baseada no protocolo SNMP (*Simple Network Management Protocol*) para a gerência e são considerados preponderantemente os padrões para controle de consumo dos equipamentos *Advanced Power Management*, APM, e *Advanced Configuration and Power Interface Specification*, ACPI. Além da arquitetura do gerente e dos agentes, foi definida também uma MIB (*Management Information Base*) para controle de consumo.

No projeto do sistema, foi privilegiado o objetivo de utilização em qualquer ambiente de rede, sem preferência por equipamentos de algum fabricante específico ou por arquitetura de *hardware*. Tecnologias de domínio público foram utilizadas, quando possível. No futuro este sistema pode fazer parte da distribuição de sistemas operacionais, incorporando controle de consumo às redes.

No texto é feita uma comparação entre os *softwares* existentes para controle de consumo, são apresentados os recursos de controle de consumo disponíveis nos equipamentos de computação, seguido da descrição do protocolo de gerência utilizado. Em seguida, é apresentada a proposta detalhada do sistema de controle e descrita a implementação do protótipo.

Palavras-chave: controle de consumo, SNMP, UPS, MIB, gerência de redes, tolerância a falhas.

Title: A Power Control System for Computer Networks

Abstract

In this work, we propose and describe the implementation of a power control system for computer networks, which aims at both increasing the operation time of the network during power failures and also to reduce power consumption during normal operation.

The definition of NetPower system, as it has been called, is based on a structure coordinated by a manager that monitors network equipment activity. It defines consumption states changing to satisfy to power restrictions and optimization goals. It is possible to establish a hierarchical approach to equipment organization with particular behavior for different groups or isolated components. A backup manager has been defined as an option to the manager failures.

The technologies used are the SNMP - Simple Network Management Protocol for management and the standards for equipment power control Advanced Power Management, APM, and Advanced Configuration and Power Interface Specification, ACPI. Besides the definition of the manager/agents architecture, we also defined a power control Management Information Base, MIB.

The NetPower system design envisages a nonrestrictive use interacting with different kind of equipment, manufacturers and variable management platforms. We just use non-proprietary and public domain technologies. In the future, this system may be part of operating systems distribution, adding power control to the networks.

In this text, we present first a comparison among some commercially available software for power control and after the resources available for power control in the equipment. A brief description of the management protocol finishes the conceptual part. Afterwards, we present the architecture of the system in details and the description of the implemented prototype.

Keywords: power control, SNMP, UPS, MIB, network management, fault tolerance.

1 Introdução

O controle do consumo de energia vem despertando atenção crescente em vários segmentos da sociedade. Primeiramente, porque equipamentos que operam com fonte limitada de energia (baterias) estão se tornando bastante difundidos. Além disso, o controle de consumo permite a redução do custo de operação dos equipamentos e a economia de recursos naturais para a geração de energia, que são limitados. Antigamente, o enfoque de um grande computador por instalação (*mainframe*) tinha um impacto muito diferente sobre o consumo energético do que as redes de computadores pessoais (cada um com seu vídeo) - que proliferam hoje nas instituições.

Equipamentos de computação, especificamente, dependem do fornecimento de energia de boa qualidade para seu funcionamento, o que não pode ser garantido apenas ligando-se o equipamento diretamente à rede elétrica da concessionária. Alteração na qualidade referente a algumas características e mesmo falta de energia podem ocorrer a qualquer instante e causar perda de dados ou danos aos equipamentos.

Em ambientes que executam aplicações confiáveis, equipamentos genericamente chamados *Uninterruptible Power Supply* - *UPS* - são interligados entre os equipamentos de computação e o ponto de fornecimento da concessionária. Possuem como objetivos: fornecer energia em caso de falha no fornecimento da concessionária e garantir que a qualidade da energia fornecida seja adequada às exigências dos equipamentos. Equipamentos UPS também são coloquialmente conhecidos por *nobreaks*.

Existem UPS de vários tipos e algumas podem ser monitoradas e gerenciadas remotamente. A gerência da UPS permite efetuar ações sobre a UPS, tais como disparar procedimentos remotos de teste, rotinas de diagnóstico, ou mesmo alterar o estado de funcionamento da UPS, incluindo seu desligamento. A partir do monitoramento, é possível detectar falhas no fornecimento de energia e executar ações que minimizem os prejuízos advindos dessas anomalias. As ações executadas, via de regra, visam salvar o contexto do sistema computacional, preparando-o para o desligamento, caso a falha de fornecimento seja prolongada.

Entretanto, em caso de falha no fornecimento da concessionária, é possível prolongar o tempo em que a UPS sustenta os equipamentos da rede com carga das suas baterias, através da racionalização do consumo. Equipamentos que não sejam essenciais para o funcionamento da rede podem ser desligados ou colocados em estados de baixo consumo. Equipamentos que sejam essenciais para que a rede continue oferecendo serviços, podem ter subsistemas ociosos desligados, como o monitor, por exemplo.

Paralelamente a esta necessidade operacional, existem outros aspectos práticos, notadamente a forma pela qual são usados os computadores pessoais. O uso individualizado das máquinas faz com que boa parte apresente taxas altas de inatividade - ao mesmo tempo que se recomenda que elas não sejam ligadas e desligadas várias vezes por dia, por exemplo. A impaciência dos usuários para a inicialização do sistema poderia ser identificada como um dos fatores que contribuí ao atendimento desta recomendação.

Assim, aos poucos, começavam a ser introduzidos mecanismos que preservam a vida útil dos equipamentos (notadamente na proteção de vídeo) e que reduzem o consumo elétrico, sem implicar no desligamento propriamente dito.

Hoje, alguns equipamentos de computação possuem mecanismos para controle de consumo embutidos, sendo necessário apenas o desenvolvimento de *software* que interaja com as interfaces desses mecanismos. Para os computadores do tipo PC (*Personal Computer*), existem as especificações APM (*Advanced Power Management*) e ACPI (*Advanced Configuration and Power Interface Specification*) que padronizam o controle de consumo e definem a interface para que o sistema operacional (SO) e as aplicações interajam com o *hardware*. Equipamentos que não possuam esses mecanismos necessitam *hardware* adicional para o controle.

Segundo um estudo desenvolvido nos Estados Unidos da América [NOR96], computadores pessoais que possuam mecanismos de controle de consumo podem economizar entre 40 kWh e 200 kWh por ano. O estudo sugere que a economia pode ser de US\$ 3,00 a US\$15,00¹ por computador. Outro dado apresentado por este estudo informa que, para cada 4 KW economizados no consumo dos equipamentos, 1 KW é economizado em refrigeração do ambiente.

Vale lembrar, entretanto, que o uso de mecanismos tais como os definidos pelas especificações APM e ACPI dependem de configurações feitas pelo próprio usuário (ou por quem instalou o sistema). Esta atitude (definição pelo usuário) será referida aqui como política local.

Em situações de fornecimento normal, é possível determinar políticas de controle de consumo que complementem as políticas locais, obtendo-se economia de recursos financeiros. Por exemplo, alguns equipamentos podem ser desligados à noite ou durante finais de semana, de acordo com as determinações do administrador da rede ou em consonância com as informações sobre a situação do sistema de energia.

Este trabalho propõe um sistema de controle de consumo que possa ser facilmente adaptado para funcionar com qualquer tipo de UPS e que atinja os seguintes objetivos:

- estenda o tempo de duração das baterias ao máximo possível, desligando subsistemas não vitais dos equipamentos;
- utilize o suporte à gerência de energia existente nos equipamentos, quando possível;
- permita gerenciar equipamentos que não possuem suporte à gerência de energia;
- possa ser utilizado mesmo quando não há falhas de fornecimento, objetivando economia de recursos financeiros.

¹ O estudo adota o preço de US\$ 0,0775 para o kWh. No Rio Grande do Sul, Brasil, o valor médio em 1999 foi de R\$ 0,09595 o kWh [ANE2000].

Para atingir estes objetivos, é necessário um modelo de gerência adequado. O modelo de gerência na Internet, baseado no protocolo SNMP (*Simple Network Management Protocol*) foi escolhido por possuir diversas características interessantes:

- é um protocolo fixado por agências de padronização, não sendo de propriedade de fabricantes isolados;
- é um padrão bastante difundido e implementado em grande variedade de equipamentos;
- objetiva consumir a menor quantidade de recursos possível para a gerência;
- é de fácil implementação.

O modelo básico de gerência na Internet é composto de um módulo gerente, que reúne informações sobre os equipamentos gerenciados e atua nesses equipamentos. Os componentes de hierarquia inferior ao gerente correspondem aos equipamentos, que enviam informações ao gerente e executam localmente as ações determinadas por este. No âmbito do SNMP, esta hierarquia é implementada através de gerente e agentes. Assim, para a gerência de consumo, deve ser desenvolvido um gerente e diversos agentes, um para cada tipo de equipamento a ser controlado.

O sistema proposto neste trabalho, e que recebeu a denominação de *Network Power Control System*, NetPower, é mostrado na figura 1.1; utiliza um *software* gerente responsável por combinar as informações da UPS, dos agentes dos equipamentos e as determinações do administrador para executar as medidas de controle de consumo adequadas; *softwares* agentes que residem nos equipamentos e realizam a interface entre o gerente e o equipamento, informando ao gerente mudanças no estado de consumo do equipamento e recebendo do gerente instruções para a execução de ações; a UPS informa ao gerente a ocorrência de eventos relacionados à alimentação, controla a qualidade da energia entregue aos equipamentos e utiliza baterias para alimentá-los em caso de falha na alimentação convencional; o gerente reserva deve substituir o gerente principal em caso de falha deste; uma interface pode ser ligada ao gerente para que o administrador possa visualizar o estado de consumo dos equipamentos da rede.

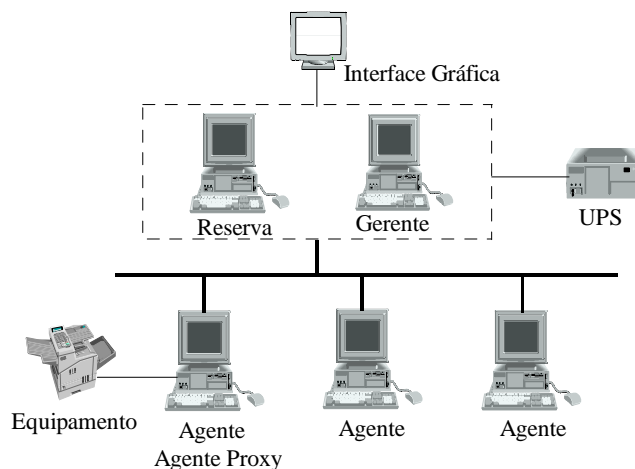


FIGURA 1.1 - Sistema proposto

Existem *softwares* desenvolvidos por fabricantes de UPS, vendidos junto com estes equipamentos, que realizam as funções de controle e monitoração dos

equipamentos mas apresentam diversas limitações, como por exemplo, são desenvolvidos especificamente para a UPS do fabricante, alguns atuam apenas no equipamento que monitora a UPS, possuem capacidade limitada de detecção de falhas no fornecimento e de atuação nos equipamentos nem permitem a criação de hierarquia para o fornecimento de energia quando o consumo deve ser racionalizado.

O sistema NetPower apresenta as seguintes contribuições:

- a proposta de um sistema de uso geral, independente do modelo de UPS e dos equipamentos gerenciados;
- a utilização de estados de baixo consumo nos equipamentos da rede;
- a gerência de subsistemas ociosos dos equipamentos da rede;
- um sistema que possa se adaptar à evolução tecnológica atuando de acordo com as novas estratégias de economia de energia.

O trabalho desenvolvido encontra-se inserido em um projeto de parceria entre o Instituto de Informática da Universidade Federal do Rio Grande do Sul e da empresa CP Eletrônica S.A., chamado *Sistema de Controle Confiável: uma aplicação em energia* [JAN97], que, entre outros, possui o objetivo de desenvolver um sistema de controle de consumo para ambientes de redes de computadores alimentados por equipamentos UPS.

Os próximos capítulos apresentam conceitos necessários à compreensão dos tópicos apresentados acima: o capítulo 2 apresenta os equipamentos UPS, suas características básicas e as soluções de *software* existentes para realizar a gerência de energia; o capítulo 3 descreve o protocolo SNMP; o capítulo 4 sintetiza as características de suporte à gerência de energia encontradas atualmente nos equipamentos de computação. Os demais capítulos descrevem a proposta do sistema de gerência: o capítulo 5 apresenta o sistema de gerência proposto; o capítulo 6 apresenta o principal componente do sistema, o gerente NetPower; o capítulo 7 descreve o protótipo implementado para validação do sistema. Por fim, o capítulo 8 apresenta as considerações finais e possíveis trabalhos futuros neste tema.

2 Equipamentos UPS e Softwares Relacionados

Os equipamentos da classe genericamente denominada *Uninterruptible Power Supply*, UPS, possuem circuitos eletrônicos que monitoram constantemente a qualidade da energia entregue pela concessionária, corrigindo eventuais distorções que possam ser malélicas a equipamentos de computação. Além disso, possuem baterias para fornecer energia em caso de interrupção no fornecimento da concessionária. A energia das baterias é utilizada com dois propósitos: prover energia para equipamentos cujo funcionamento seja importante para a disponibilidade dos serviços da rede e permitir que os equipamentos menos importantes possam ser colocados em estado seguro (contexto salvo, arquivos fechados) antes do corte no fornecimento. As próximas seções apresentam, inicialmente, as formas básicas de funcionamento das UPS e o modelo de comunicação adotado neste trabalho para as UPS trocarem informações. Logo após, são apresentadas algumas características operacionais das UPS encontradas no mercado, uma análise crítica destas e as características funcionais do *software* proposto.

2.1 Classificação das UPS

As UPS podem ser classificadas de acordo a maneira com que a energia é fornecida à carga. Existem três tipos: *online*, *standby* e híbrida, conforme a figura 2.1 [PER00].

As UPS do tipo *standby* repassam energia da concessionária para os equipamentos. Ao detectar interrupção no fornecimento da concessionária, passam a fornecer energia das baterias até o seu esgotamento ou até ser restabelecido o fornecimento. A desvantagem deste tipo de UPS é o tempo necessário para ativar o fornecimento das baterias em caso de falha no fornecimento comercial. Esta dificuldade de transferência acentua-se drasticamente em equipamentos de potência elevada. As UPS do tipo *online* sempre fornecem energia das baterias, utilizando a energia comercial apenas para carregá-las. Neste caso, não há atraso para comutação, mas a vida útil das baterias é reduzida, se comparada aos parâmetros registrados em equipamentos do tipo *standby*. As UPS híbridas fornecem energia das baterias, como as UPS *online*, mas ao detectarem algum problema (como sobrecarga ou falha no inversor, por exemplo), transferem a energia recebida da fonte externa através do *bypass*.

Um aspecto importante refere-se à capacidade de gerência, e característica que permite a identificação de três tipos de UPS. As mais simples não permitem gerência e fornecem energia até o esgotamento completo das baterias. Alguns tipos chamados de UPS de contato seco possuem interface de comunicação (geralmente através de porta serial) e informam a ocorrência de alguns eventos a um computador conectado. Este tipo de UPS pode ser utilizado de maneira limitada no controle de consumo. Existem ainda UPS chamadas de inteligentes que permitem a gerência completa de diversos fatores que afetam o fornecimento de energia e permitem controle completo da energia fornecida.

O trabalho de Peres [PER00] descreve de forma abrangente os tipos de UPS existentes e suas principais características. Também descreve a gerência destes equipamentos através do protocolo SNMP.

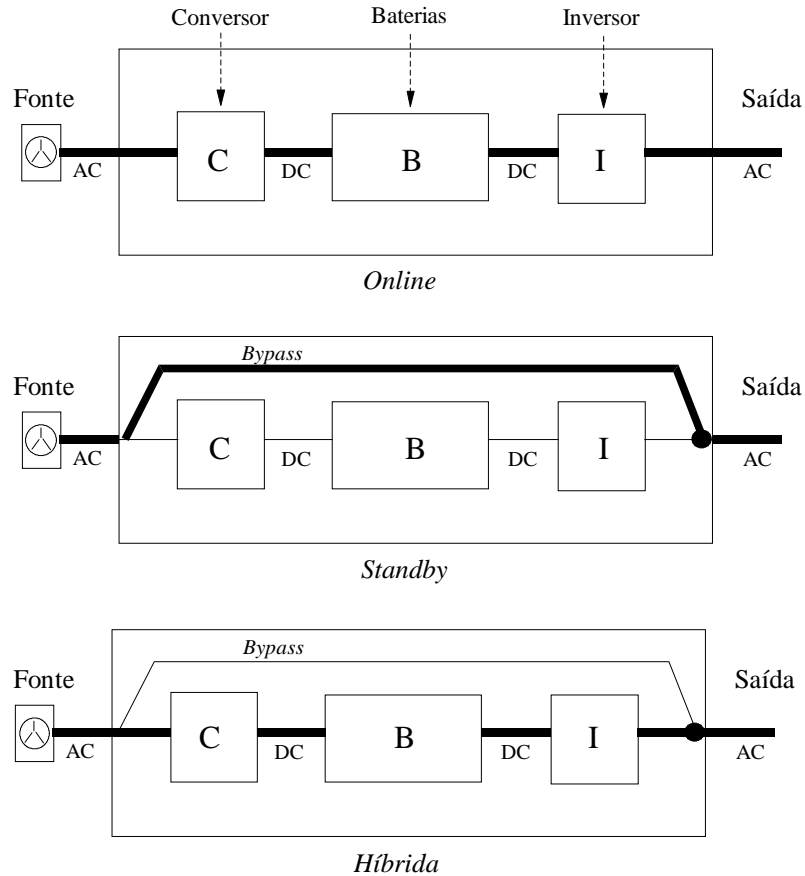


FIGURA 2.1 - Classificação das UPS quanto à forma de fornecimento de energia

As UPS podem ser empregadas basicamente de três maneiras. A primeira é ligar à UPS apenas os equipamentos mais importantes da rede, como servidores por exemplo, para aumentar sua disponibilidade e evitar danos aos dados. As estações clientes e outros equipamentos (impressora, hubs) são desligados imediatamente, em caso de falta de energia. Esta abordagem possui custo reduzido, mas causa transtornos aos usuários da rede, podendo causar perda de dados nos equipamentos clientes.

A segunda maneira, é ligar todos os equipamentos da rede à UPS. Dessa maneira, todos os equipamentos da rede são protegidos. No entanto, o custo desta solução é mais alto e o tempo de duração da carga das baterias é menor. O sistema proposto é projetado para este tipo de ambiente.

A terceira maneira de se utilizar as UPS, que na verdade é complementar às duas anteriores, é como fonte intermediária de fornecimento, por tempo suficiente para que um gerador seja acionado. As vantagens do gerador são o custo reduzido do combustível e a virtual ausência de limite no tempo de fornecimento. Neste caso, a UPS deve fornecer energia por apenas alguns minutos, até que a capacidade de fornecimento do gerador seja estabilizada. O sistema proposto também possui utilidade neste tipo de

ambiente, colocando os equipamentos em estado de baixo consumo até que o gerador seja acionado.

2.2 Conexão das UPS à Rede

Para este trabalho, assume-se que a UPS possua alguma interface de comunicação com a rede. Geralmente utiliza-se o protocolo SNMP para a gerência de UPS. Nesse caso, a UPS pode ser gerenciada de duas maneiras, conforme a figura 2.2.

No primeiro caso (figura 2.2a), a UPS possui uma camada de *software* que implementa o protocolo SNMP e pode ser ligada diretamente à rede. As mensagens do SNMP são enviadas diretamente à UPS.

No segundo caso (figura 2.2b), a UPS possui alguma interface de comunicação (geralmente comunicação serial) e algum protocolo para comunicação com um *software* de controle. Nesse caso, deve ser desenvolvido um agente chamado *proxy* (seção 3.2) que realiza o interfaceamento entre o protocolo da UPS e o SNMP e a conecta à rede. Nesse caso, o gerente comunica-se com o agente *proxy* e este transfere os dados à UPS.

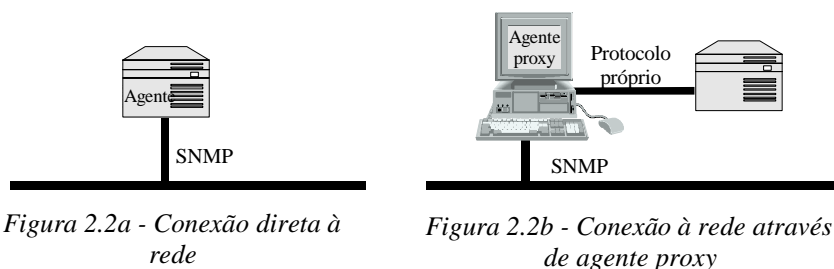


FIGURA 2.2 - Conexão da UPS à rede

2.3 Softwares Comerciais de Controle de Consumo

As UPS comercializadas são geralmente acompanhadas de algum *software* de controle desenvolvido pelo próprio fabricante da UPS. Esses pacotes de *software* destinam-se à configuração da UPS e alguns permitem controlar a utilização da energia das baterias. Neste trabalho, são avaliados apenas os aspectos relativos à utilização da energia das baterias e não à configuração da UPS.

Foram avaliados os *softwares* de alguns fabricantes que, acredita-se, sejam representativos das opções disponíveis no mercado. Estes *softwares* foram obtidos pela Internet, nas páginas dos fabricantes. Os fabricantes e os *softwares* são listados na tabela 2.1 e são analisados nas seções seguintes.

TABELA 2.1 - Softwares analisados e seus fabricantes

<i>Fabricante</i>	<i>Software</i>
Liebert Corporation	SiteNet Multilink
American Power Conversion	PowerChute

<i>Fabricante</i>	<i>Software</i>
Engetron	SUPSNNet

2.3.1 Software SiteNet Multilink

O *software* SiteNet Multilink é vendido junto com as UPS da empresa Liebert Corporation. [LIE99]. Permite atuar em computadores isolados ou em rede. Durante a instalação, o usuário indica se o equipamento é o monitor da UPS ou um cliente. Caso seja o monitor, deve-se indicar em qual porta serial a UPS encontra-se conectada e os computadores-clientes que receberão as notificações de eventos da UPS.

Após detectar falha no fornecimento, o *software* notifica os computadores clientes e dispara temporizadores. Cada um dos temporizadores que atingir o tempo programado antes do retorno da energia, executa *scripts* para encerrar as aplicações e o sistema operacional dos equipamentos. Se a energia retornar, os temporizadores são desativados.

Pode ser utilizado nos seguintes cenários:

- um computador ligado a uma UPS,
- uma rede com vários computadores ligados a uma UPS,
- diversas redes interconectadas alimentadas por uma ou mais UPS,

e através dos seguintes sistemas operacionais: Microsoft Windows 95/98/NT², Sun Solaris³, Hewlett Packard HP-UX⁴, IBM AIX⁵, SCO OpenServer⁶, Novell Netware⁷ e OS/2 Warp⁵.

Possui interface e funcionalidade bastante simples. A figura 2.3 apresenta a tela de configuração das ações a serem executadas na ocorrência de eventos. Essas ações devem ser configuradas em cada computador gerenciado. Podem ser programados três tipos de ações:

- notificação ao usuário - mostra caixa de diálogo informando eventos ao usuário;
- execução de *scripts* - permite que o administrador indique *scripts* a serem executados na ocorrência de eventos e quanto tempo o sistema deve aguardar antes de executá-los. Os eventos monitorados são:
 - *on_battery* - quando a UPS passa a fornecer energia ao sistema;
 - *low_battery* - quando a carga das baterias atinge nível baixo;
 - *return_to_normal* - quando o fornecimento retorna ao normal;

² Microsoft Corporation

³ Sun Microsystems

⁴ Hewlett-Packard Company

⁵ IBM Corporation

⁶ Santa Cruz Operation Corporation

⁷ Novell Corporation

- encerramento do sistema operacional - permite estabelecer o tempo de espera antes de encerrar o sistema operacional para os eventos *on_battery* e *low_battery*.

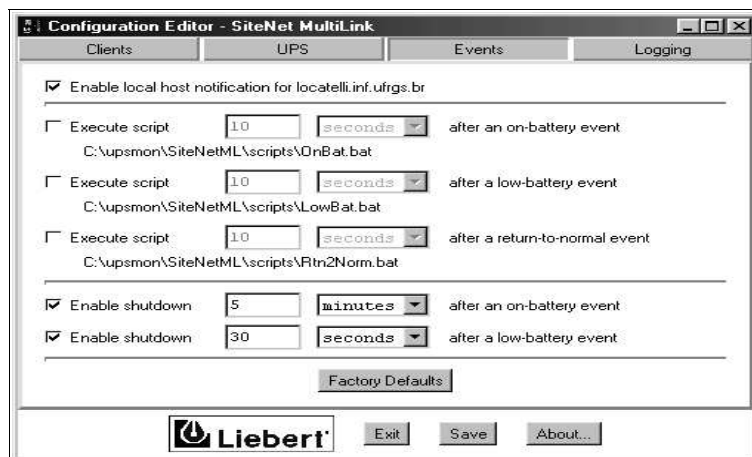


FIGURA 2.3 - Tela de configuração de ações do SiteNet Multilink

Assim, através de configuração adequada, a execução de ações no sistema permite o acompanhamento da situação de carga das baterias da UPS, que pode ser: *normal*, *on battery*, *low battery* ou *weak battery*; permite a geração de *log* da ocorrência de eventos e ainda que o usuário cancele o encerramento do sistema operacional determinado pelo monitor da UPS.

2.3.2 Software PowerChute

O software PowerChute da American Power Conversion ou APC Corporation () [APC99] é fornecido junto com as UPS desta empresa. Realiza as funções de monitorar a UPS e atuar em função da ocorrência de eventos.

Pode ser integrado com a plataforma de gerência da Compaq, o CIM (*Compaq Insight Manager*), e com o SMS (*System Management Server*) da Microsoft.

O sistema consiste de dois módulos: um de controle da UPS e um de interface com o usuário. A figura 2.4 apresenta a tela principal do programa.

Ainda de acordo com o constante nas informações comerciais, o sistema:

- permite o encerramento do sistema operacional e de alguns *softwares*, como BackOffice (SQL Server, SMS, SNA Server, IIS)⁸, Netscape Web Server⁹, Lotus Notes¹⁰;
- permite programação de ações com data marcada. Estas ações podem ser: auto-teste da UPS, calibração da UPS ou encerramento do sistema operacional. Estes eventos podem ser programados para ocorrer em determinada data e horário, em determinado dia da semana ou do mês;

8 Microsoft Corporation

9 Netscape Corporation

10 IBM Corporation

- monitora 48 eventos da UPS que podem disparar ações. As ações podem ser: gravação de *log*, envio de notificação ao administrador, envio de notificação ao usuário, execução de *script*, envio de *email*, envio de sinal de *pager* ou encerramento do sistema operacional. A figura 2.5 mostra a janela de configuração de ações.

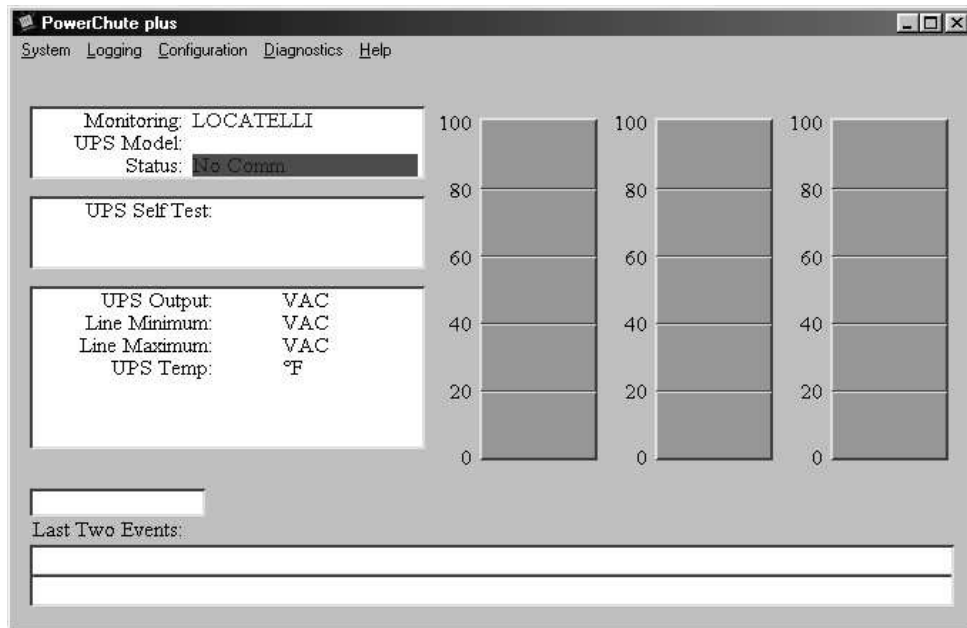


FIGURA 2.4 - Tela principal do PowerChute

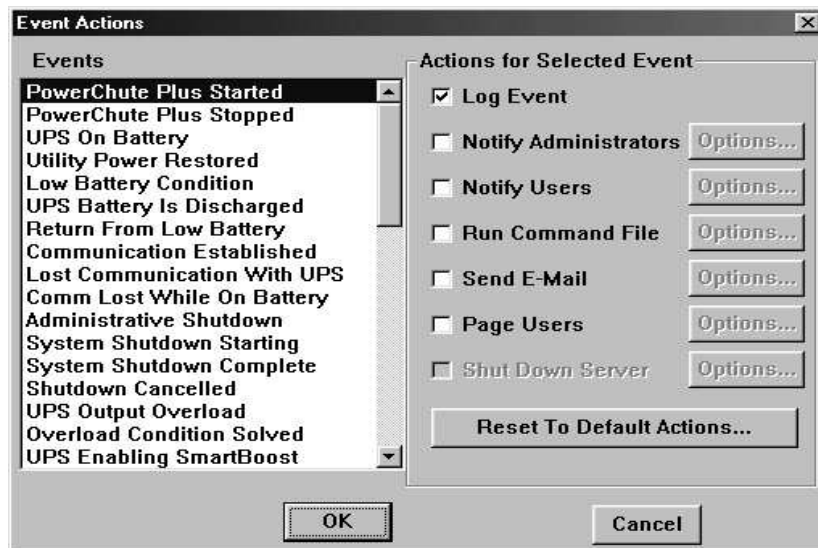


FIGURA 2.5 - Tela de configuração de ações do PowerChute

As ações de gravação de *log*, execução de *script*, encerramento do sistema operacional ocorrem apenas nos computadores que possuem o *software* instalado; portanto ele deve ser instalado e configurado em todos os computadores da rede que devam ser monitorados.

2.3.3 Software SUPSNet

O SUPSNet é o *software* que acompanha as UPS da empresa Engetron e realiza a supervisão da UPS e atuação nos equipamentos, quando necessário [ENG99]. Pode ser executado com os sistemas operacionais Windows 95 ou NT.

É dividido em módulos e possui um servidor Telnet, um servidor HTTP e um agente *proxy* SNMP, que permitem a gerência da UPS da maneira mais conveniente ao administrador. Os servidores Telnet e HTTP são do tipo monousuário, e exigem senhas para a realização de operações.

O agente SNMP implementa a MIB padrão para gerência de UPS definida pela RFC 1628 [CAS94]. Os objetos implementados são: **upsIdent**, **upsBattery**, **upsInput**, **upsOutput**, **upsAlarm**, **upsTest**, **upsControl**, **upsConfig** e **upsTraps**. O agente também pode enviar interrupções, se for configurado. A documentação não menciona a versão do SNMP, mas não foram identificadas características do SNMPv3.

O *software* monitora as seguintes variáveis internas da UPS: tensão, corrente e frequência para entrada, saída e bateria; autonomia da UPS em minutos; temperatura interna; indicadores de sobrecarga e outros.

O *software* permite estabelecer horários para que a UPS ative as saídas de energia. Permite a execução de três tipos de ações: envio de mensagem de advertência, encerramento do sistema operacional local e execução de programa externo (*script*).

O envio de mensagem pode ser utilizado apenas em Windows NT. O encerramento pode ser feito em Windows 95 ou NT, na máquina local. Para o encerramento de máquinas remotas, é fornecido um *software* que deve ser executado a partir de um arquivo de *script*.

2.4 Considerações sobre os Softwares Estudados

Os *softwares* apresentados nas seções anteriores possuem diversas características comuns:

- funcionam apenas com as UPS do fabricante para os quais foram desenvolvidos;
- não gerenciam dispositivos internos aos equipamentos;
- não utilizam estados de consumo reduzido;
- alguns atuam apenas no computador que está conectado à UPS;
- não permitem tratar os equipamentos de maneira diferenciada;
- possuem número limitado de eventos;
- geram *log* apenas da ocorrência de eventos, não da execução de ações;
- possuem interface gráfica apenas para acompanhamento da UPS, e não dos equipamentos gerenciados.

Nenhum dos *softwares* analisados se propõe a interagir com UPS de outros fabricantes. Portanto, em ambientes com UPS de diferentes fabricantes, devem ser instalados *softwares* compatíveis com cada uma. A análise dos *softwares* mostrou que esses apresentam bastante diversidade em diversos aspectos, como objetivo, organização, apresentação, configuração e funcionalidade. Em um ambiente heterogêneo, devem ser utilizadas aplicações de gerência que facilitem a interoperabilidade dos equipamentos.

Nenhum *software* analisado permite o desligamento de subsistemas dos equipamentos. Entretanto, em muitas situações de uso, subsistemas como o monitor, por exemplo, podem ser desligados sem prejuízo para o desempenho das funções dos equipamentos.

Nenhum dos *softwares* permite colocar o equipamento em estado de consumo reduzido. Estados de consumo reduzido são úteis por economizar energia e permitir que o equipamento retorne ao estado de atividade normal em pequeno espaço de tempo.

Os *softwares* realizam o mesmo tipo de ações em todos os equipamentos componentes da rede gerenciada. Certamente alguns equipamentos da rede devem ser favorecidos em situações de alimentação crítica, havendo a necessidade de estabelecer hierarquia no fornecimento de energia.

Todos os *softwares* possuem um conjunto fixo de eventos a serem monitorados, ou seja, não é possível redefinir o conjunto de eventos monitorados para adaptação ao ambiente da rede que será alvo da gerência. O SiteNet Multilink monitora três eventos. O PowerChute monitora 48 eventos. A documentação do SUPSNet não especifica quantos eventos são monitorados, mas tampouco permite alteração destes.

Todos os *softwares* geram *logs* apenas da ocorrência de eventos, mas não das ações executadas. O *log* das ações seria importante para a identificação de falhas na gerência.

Nenhum dos *softwares* analisados possui interface gráfica para acompanhar o estado de consumo dos equipamentos da rede. Esta interface seria interessante para que o administrador humano pudesse identificar situações passíveis de economia e atuar nos equipamentos de maneira específica.

2.5 Características do Sistema Proposto

O sistema proposto neste trabalho apresenta a soma das características consideradas interessantes, encontradas nos *softwares* analisados, e algumas outras que são importantes mas não foram encontradas nos sistemas comerciais. A tabela 2.2 apresenta as características adotadas, incluindo tanto as encontradas nos *softwares* analisados, como as introduzidas pelo sistema proposto.

TABELA 2.2 - Características dos softwares de gerência

<i>Característica</i>	<i>SiteNet Multilink</i>	<i>PowerChute</i>	<i>SupNet</i>	<i>Sistema proposto</i>
Monitoração de eventos	sim	sim	sim	sim
Execução de ações	sim	sim	sim	sim
Envio de mensagens	sim	sim	sim	sim
Execução de <i>scripts</i>	sim	sim	sim	sim
Encerramento do sistema operacional	sim	sim	sim	sim
Geração de <i>log de eventos</i>	sim	sim	sim	sim
Integração com plataforma de gerência	não	sim	não	sim
Interação do usuário com o sistema	sim	não	não	sim
Execução de ações com hora marcada	sim	sim	não	sim
Interoperabilidade	não	não	não	sim
Estados de consumo reduzido	não	não	não	sim
Ligação automática	não	não	não	sim
Gerência de equipamentos diversos	não	não	não	sim
Gerência de dispositivos	não	não	não	sim
Configuração dos eventos monitorados	não	não	não	sim
Modularização	não	não	não	sim
Gerência durante operação normal	não	não	não	sim

Todos os sistemas são orientados a eventos, ou seja, aguardam que a UPS sinalize uma mudança no estado de energização para a adoção de medidas. Igualmente, todos os sistemas permitem a execução de ações em função das notificações de eventos pré-definidos. Essas ações consistem no envio de mensagens, na execução de *scripts* ou programas externos e no encerramento do sistema operacional. Adicionalmente todas realizam registro da ocorrência de eventos em arquivos de *log*.

Alguns *softwares* permitem adicionalmente a integração com plataformas de gerência (apresentadas na seção 5.10), a interação de usuários para que estes possam solicitar maior tempo de uso do equipamento antes do desligamento e a execução de ações com hora marcada.

As últimas linhas da tabela 3.2 contêm características originais do sistema proposto: o modelo adotado e a solução de implementação destas constitui uma das

contribuições deste trabalho. Estas características são explicadas a seguir, como entendidas neste trabalho.

A interoperabilidade permite que UPS de fabricantes diferentes sejam utilizadas de maneira transparente. A interface com o administrador humano e a configuração do *software* são realizadas sempre da mesma maneira. É necessário apenas escrever um pequeno módulo de comunicação com a UPS.

A utilização de estados de consumo reduzido permite que se obtenha economia de recursos, mas que o equipamento ainda esteja disponível para uso em curto período de tempo. Equipamentos que possam ser utilizados remotamente e equipamentos de utilização esporádica mas de funcionalidade essencial podem se beneficiar desta característica.

Diferentemente dos *softwares* estudados, o sistema NetPower permite que os equipamentos sejam ligados automaticamente pelo gerente em caso de retorno do fornecimento normal de energia ou em horário programado.

O sistema NetPower também objetiva a gerência de outros equipamentos além de computadores. Dessa maneira, impressoras, *hubs*, luminárias e quaisquer outros equipamentos elétricos podem ter seu consumo controlado, desde que o devido *hardware* de controle seja providenciado.

A gerência de dispositivos permite que subsistemas ociosos sejam desligados. Assim, um equipamento não precisa estar completamente energizado para desempenhar suas funções.

O sistema proposto permite que o administrador monitore os eventos que achar necessários das UPS. Se a necessidade de gerência for simples, pode-se definir apenas alguns eventos a serem monitorados (como falha e retorno do fornecimento, por exemplo). Se o cenário de gerência for complexo, podem ser monitorados eventos que detalham a situação do fornecimento de energia (como falhas no fornecimento, nível de carga baixo, nível de carga crítico e retorno do fornecimento, por exemplo). A definição dos eventos envolve a programação do respectivo módulo de comunicação com a UPS.

A modularização do *software* gerente é a característica que permite interoperabilidade, integração com plataformas de gerência e funcionamento em segundo plano. As soluções encontradas implementam todo o *software* de gerência em um único módulo.

A infra-estrutura de gerência também é aproveitada para permitir a gerência de consumo durante fornecimento normal de energia, possibilitando economia de recursos financeiros. Por exemplo, o gerente pode ser programado para desligar alguns equipamentos durante o final de semana ou em períodos de ociosidade não intencional e excessivamente prolongada. Isto é particularmente interessante em instalações com grande quantidade de pontos individuais de trabalho (tais como estações de trabalho e computadores do tipo pessoal) para os quais haja possibilidade de acesso remoto, ou onde não tenha sido implantada uma política disciplinada de desligamento.

3 O Protocolo SNMP

O protocolo *Simple Network Management Protocol*, SNMP, é hoje o padrão de gerência para redes de computadores [STA96, ROS91, PER00]. Além do próprio protocolo de comunicação, o modelo de gerência SNMP define uma plataforma de gerência, geralmente referenciada como plataforma de gerência na Internet.

A origem do SNMP está relacionada com o crescimento do uso da Internet e a ausência de meios de gerência para essa rede. Originalmente não havia protocolo de gerência e utilizava-se o ICMP (*Internet Control Message Protocol*), um protocolo de comunicação entre roteadores, para identificar equipamentos que não estivessem funcionando [STA96].

Durante a década de 1980, quando a Internet passou a crescer de maneira exponencial, os organismos reguladores, especialmente a IAB (*Internet Architecture Board*), passaram a estudar propostas de protocolos de gerência.

Inicialmente o SNMP era visto como um elemento de transição para o protocolo proposto pela OSI (*Open Systems Interconnection*), o CMOT (*Common Management Information Protocol over TCP/IP*), por ser este último considerado de implementação difícil. Foi previsto que posteriormente o SNMP evoluiria em direção ao CMIP, o que não aconteceu. Atualmente as duas plataformas evoluem independentemente, sendo o CMIP utilizado largamente na gerência de redes de telecomunicações e o SNMP, em redes de computadores [STA96].

O SNMP tem sofrido adaptações desde sua padronização. A terceira versão foi padronizada recentemente e espera-se que seja a última versão do protocolo. Todas as versões (SNMPv1, SNMPv2c, SNMPv3) apresentam elevado grau de compatibilidade e este capítulo as trata separadamente apenas quando necessário.

É necessário esclarecer que a segunda versão do protocolo teve seu desenvolvimento conturbado por divergências entre o grupo de padronização e acabou gerando diversas subversões. Dessas, a mais difundida, e que se tornou a base do SNMPv3, é a versão SNMPv2c.

O modelo de gerência na Internet é composto de quatro elementos: **gerente**, **agente**, **base de informações de gerência** e **protocolo de gerência**, conforme a figura 3.1. Os gerentes e agentes também são chamados **entidades do SNMP**. Ao longo deste capítulo, são apresentadas estas entidades e o funcionamento básico do protocolo, incluindo as estruturas de apoio empregadas, como a base de informações. Ao final, são comentados aspectos específicos de segurança, que diferenciam fundamentalmente a versão 3 de suas antecessoras.

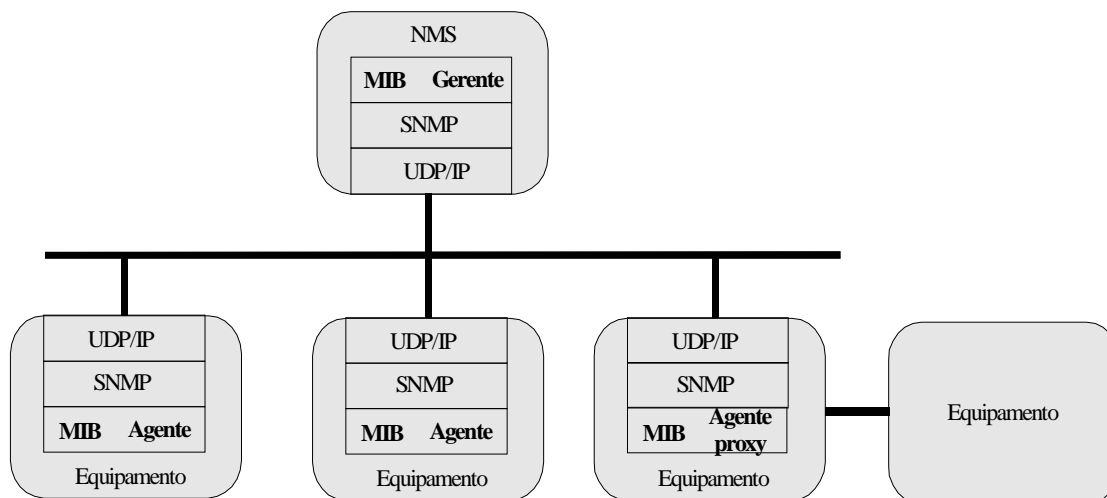


FIGURA 3.1 - Elementos do modelo de gerência na Internet

3.1 Gerente

O **gerente** é um *software* que possui três funções: capturar informações sobre os nodos gerenciados da rede, atuar nestes nodos quando necessário e estabelecer a interface entre o sistema de gerência e o administrador humano. O gerente é executado em uma estação de gerência chamada NMS (*Network Management Station*), sendo composto de diversas aplicações de gerência, cada uma especializada em controlar alguns aspectos do funcionamento da rede.

De maneira geral, sempre que possível, a sobrecarga da tarefa de gerência deve ser associada ao gerente e retirada dos agentes. Os agentes devem consumir a menor quantidade possível dos recursos dos equipamentos em que são executados.

Inicialmente, o modelo de gerência utilizava apenas um gerente e diversos agentes espalhados pela rede. A partir da versão 2, é possível a criação de hierarquia de gerentes, com um gerente principal e gerentes intermediários, originando a gerência distribuída. A versão 3 do protocolo possibilita a existência de entidades que se comportem simultaneamente como agentes e gerentes.

3.2 Agente

O **agente** é um *software* que deve ser instalado em cada nodo a ser gerenciado e realiza a interface entre o sistema de gerência e o sistema operacional do equipamento. Este agente possui duas funções: responder às requisições do gerente (enviando as informações requisitadas ou atuando no equipamento), e informar ao gerente a ocorrência de situações excepcionais, através de interrupções (*traps*).

O modelo de gerência especifica um tipo de agente especial, o **agente proxy**, utilizado para controlar equipamentos que não possuam uma implementação do SNMP. Este agente deve residir em um equipamento capaz de se comunicar por SNMP e possui,

além da interface com o SNMP, uma interface com o protocolo apropriado para comunicação com o equipamento que se deseja gerenciar, conforme esquematizado na figura 3.1.

3.3 Base de Informações de Gerência

A **base de informações de gerência**, chamada de MIB (*Management Information Base*), é uma abstração dos objetos gerenciados em um nodo, organizada em uma estrutura de árvore. A organização em árvore permite o desenvolvimento independente de cada sub-árvore e a expansão da árvore, com a adição de novos ramos, sempre que necessário. A estrutura básica da MIB pode ser vista na figura 3.2 [PER00]. O gerente atua nas **folhas** da árvore MIB, que representam os **objetos gerenciados** do equipamento.

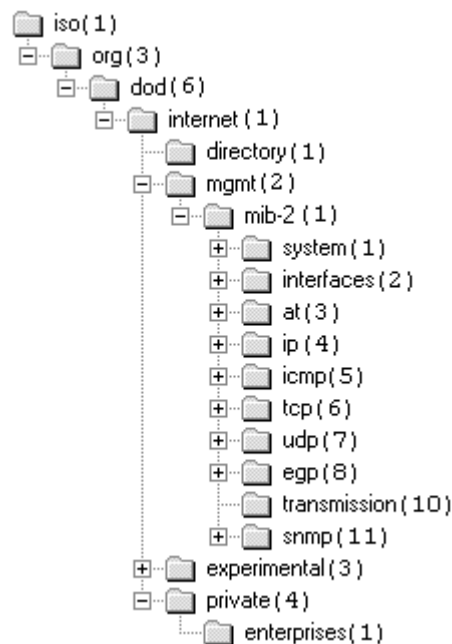


FIGURA 3.2 - Estrutura Básica da MIB

3.4 Protocolo de Gerência

O **protocolo de gerência** permite estabelecer a comunicação e a troca de informações entre o gerente e os agentes. Consiste na definição das operações disponíveis e na especificação do formato das mensagens trocadas. O protocolo também define os direitos de acesso aos dados e quem pode acessá-los.

O SNMP originalmente foi construído para operar sobre o protocolo UDP (*User Datagram Protocol*), um protocolo não orientado a conexão, que não garante entrega de mensagens nem o seu ordenamento, cabendo à aplicação realizar este controle [PER98]. A justificativa para esta escolha é que o SNMP é largamente empregado em situações extremas de tráfego da rede, possivelmente com

congestionamento e perda de pacotes, não sendo aconselhável ter por base um protocolo orientado à conexão e confiável, como o TCP (*Transport Control Protocol*). No entanto, apesar da implementação sobre UDP ser a mais comum, existem implementações de SNMP para diversos protocolos de transporte, inclusive TCP [ROS91].

3.4.1 Operações

As operações definidas para o SNMP são de quatro tipos:

- obtenção de informações - operação originada pelo gerente para monitorar os agentes. O SNMP define três operações desse tipo:

get - obtém o valor do objeto especificado;

getNext - obtém o valor do próximo objeto na árvore da MIB;

getBulk - obtém os valores de uma sub-árvore da MIB.

- alteração de informações - operação originada pelo gerente para alterar valores da MIB dos agentes, permitindo o controle dos equipamentos ligados à rede. Alterando estes valores, o gerente determina que o agente atue no equipamento. O SNMP define uma operação deste tipo:

set - altera o valor do objeto especificado.

- alerta para exceções - operação originada pelo agente para informar ao gerente a ocorrência de eventos anormais; é do tipo interrupção. Trata-se da única forma de comunicação iniciada pelo agente. Após ser informado, o gerente deve pesquisar a MIB do agente para obter os dados necessários. Existe uma operação desse tipo:

trap - informa qual exceção ocorreu.

- notificação de gerente - mensagem trocada entre gerentes. Utilizada em gerência distribuída. A operação definida é:

inform.

Todas as operações utilizam o mesmo formato de mensagem. Uma mensagem é composta de 3 campos, conforme explicado na figura 3.3 e no próximo parágrafo.

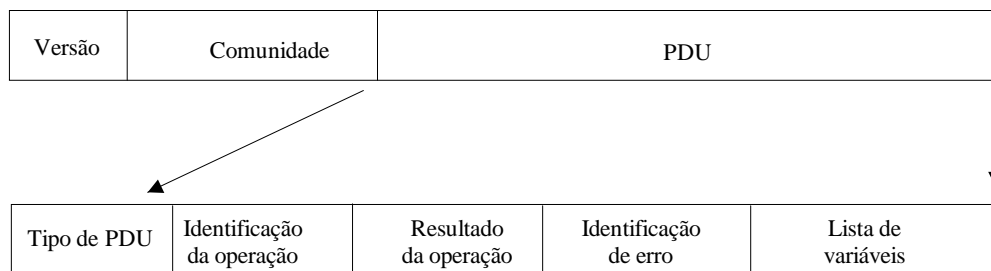


FIGURA 3.3 - Formato das mensagens do SNMP

O primeiro campo indica a versão do protocolo utilizado, o segundo indica o nome de comunidade para autenticação e o terceiro contém as operações do SNMP,

sendo chamado de PDU (*Protocol Data Unit*). Uma PDU é composta pelos sub-campos mostrados na figura 3.3. O tipo de PDU informa a operação desejada; a identificação da operação é um número que diferencia as mensagens trocadas pelas entidades; o resultado da operação indica se houve erro na operação; o sub-campo de identificação de erro indica o tipo de erro; finalmente, segue uma lista de variáveis e seus valores.

3.4.2 Segurança

Os aspectos de segurança envolvem autenticação, encriptação das mensagens e controle de acesso aos objetos da MIB. É necessário observar a versão do protocolo para a descrição de suas características de segurança.

No SNMPv1 e no SNMPv2, as mensagens trafegam como texto puro através da rede. Apenas o SNMPv3 permite encriptação das mensagens. Por esse motivo, as duas primeiras versões são vulneráveis a todos os ataques existentes contra comunicações envolvendo os protocolos UDP e IP [STA98].

Nas duas primeiras versões do protocolo, o único mecanismo de segurança corresponde a uma identificação de grupo ou nome da comunidade (*community name*), implementado através de uma cadeia de caracteres (*string*) enviada junto com a mensagem. Este identificador possui duas funções: autenticar o originador da mensagem e determinar a quais objetos este originador tem acesso. Mecanismos de segurança mais eficientes não foram implementados devido a divergências no grupo de padronização.

São associados *community names* a subconjuntos da MIB e apenas mensagens acompanhadas do identificador correto tem acesso a cada subconjunto. Assume-se que apenas usuários autorizados a lidar com determinado subconjunto devam conhecer o seu identificador de comunidade.

O uso desta forma de segurança limitou a utilização de versões anteriores do protocolo a atividades de monitoramento, havendo restrições por parte dos fabricantes à sua utilização para atuação nos equipamentos controlados através da operação *set*.

No SNMPv3, a implementação de segurança foi dividida pelo comitê de padronização em dois subsistemas: um de segurança propriamente, responsável por autenticar e encripta mensagens, e um de controle de acesso [STA98, STA98a].

Subsistema de Segurança

Para lidar com questões de segurança, o SNMPv3 utiliza o conceito de usuário (*principal*), que é a entidade que origina as operações, ao invés de comunidades. Este pode ser um indivíduo, um conjunto de indivíduos, uma aplicação, um conjunto de aplicações ou uma combinação de indivíduos e aplicações. As políticas de segurança são aplicadas em função do usuário e permitem que o administrador atribua direitos a cada usuário, de maneira individual.

O subsistema de segurança pode conter diversos modelos de segurança, mas até o momento apenas um foi padronizado, o USM (*User Security Model*). O modelo USM

permite que as mensagens sejam autenticadas e encriptadas ou apenas autenticadas. O USM foi projetado para evitar os seguintes ataques:

- alteração de informação: um espião altera uma mensagem gerada por uma entidade autorizada;
- mascaramento: uma entidade não autorizada tenta assumir a identidade de uma entidade autorizada;
- modificação do fluxo de mensagem: mensagens são reordenadas, retardadas ou repetidas para realizar operações não autorizadas;
- espionagem: uma entidade observa trocas de mensagens entre um gerente e um agente e descobre valores de objetos gerenciados e eventos.

A modificação no fluxo de mensagens pode ocorrer porque o SNMP foi projetado para funcionar sobre um protocolo de transporte sem conexão. Um dos mecanismos de defesa do SNMPv3 é o conceito de janela de tempo, que determina o tempo em que uma mensagem é válida após ser enviada.

No entanto, este modelo não prevê segurança em relação aos ataques de Negação de Serviço (*DoS - Denial of Service*) e de análise de tráfego. Não há proteção contra DoS por não ser possível distinguir um ataque de DoS de falhas na rede; e não há proteção contra análise de tráfego porque muitos padrões de tráfego são previsíveis, não havendo vantagem significativa em tentar impedir essa análise, segundo Blumenthal [BLU99]. Por exemplo, o padrão de envio de mensagens pode ser induzido a partir da verificação de que os equipamentos são gerenciados regularmente por algumas poucas estações de gerência.

Os algoritmos utilizados pelo USM são o HMAC-MD5-96 e o HMAC-SHA-96 para autenticação e o algoritmo DES (*Data Encryption Standard*) no modo CBC (*Cypher Block Chaining*) para encriptação [BLU99].

Os algoritmos de autenticação HMAC (*Hash Message Authentication Code*) [KRA97], utilizam funções de *hash* para verificar a integridade de mensagens enviadas em uma rede não confiável. As funções de *hash* utilizadas podem ser o MD5 (*Message Digest 5*) ou o SHA (*Secure Hash Algorithm*).

Subsistema de Controle de Acesso

Da mesma forma que para o subsistema de segurança, um subsistema de controle de acesso também foi definido pelo comitê de padronização.

O controle de acesso por visão (VACM – *Vision Access Control Model*), definido no RFC 2575 [WIJ99], determina se o usuário situado em um local remoto pode ter acesso aos objetos da MIB local. São definidos cinco elementos: grupo, nível de segurança, contexto, visão de MIB e política de acesso, explicados a seguir.

Um grupo é definido como um conjunto de zero ou mais pares de objetos (*securityModel, securityName*) para os quais o acesso aos objetos é permitido. O objeto *securityModel* identifica o modelo de segurança utilizado. O objeto *securityName* se

refere ao usuário; e os direitos dos usuários pertencentes ao mesmo grupo são idênticos. Cada grupo possui um nome (*groupName*) que o identifica.

O nível de segurança da mensagem pode alterar os direitos de acesso de um grupo. Por exemplo, um agente pode permitir acesso de leitura para mensagens não autenticadas mas exigir autenticação para mensagens que alterem valores da MIB.

O contexto é um subconjunto dos objetos da MIB, que podem ser agregados em coleções com diferentes políticas de acesso. Pode existir mais de um contexto em uma entidade e um objeto pode estar contido em mais de um contexto. Quando existe mais de um contexto, para a identificação de um objeto, são necessários os valores *contextName* e *contextEngineID*, além da identificação do tipo de objeto e sua instância.

O acesso aos contextos é definido através de visões de MIB (*MIB Views*), que restringem o acesso a apenas um conjunto de objetos gerenciados. Os objetos gerenciados são organizados em árvore e uma visão pode conter diversas subárvores. Uma subárvore é um nodo da MIB junto com seus elementos subordinados.

4 Controle de Consumo

Os fabricantes de computadores começaram a desenvolver mecanismos de controle de consumo apenas no início dos anos 90, de maneira independente, e primeiramente para atender às necessidades dos computadores portáteis (*notebooks*). Não existe padronização de uso generalizado para os mecanismos de controle de consumo em equipamentos de computação. No entanto, os computadores com arquitetura Intel, baseados nos processadores Pentium (referenciados apenas por PCs no decorrer do texto), possuem dois padrões que são bastante utilizados.

Este capítulo apresenta inicialmente os princípios básicos utilizados em estações de trabalho e computadores pessoais para controle de consumo e os dispositivos atualmente gerenciados. Na seqüência, são apresentados os mecanismos existentes vinculados às arquiteturas dos computadores Apple Macintosh e estações Sun Microsystems. Após, são apresentados em detalhes os mecanismos de gerência encontrados em PCs, o *Advanced Power Management*, APM, e o *Advanced Configuration and Power Management Specification*, ACPI. Ao final, são apresentadas as interfaces dos sistemas operacionais.

4.1 O Programa Energy Star

Existe um programa bastante difundido na indústria, cujo objetivo é reduzir o consumo de equipamentos durante períodos de inatividade. É o programa *Energy Star* [ENE98] do governo dos Estados Unidos da América. Este programa determina limites de consumo para classes de equipamentos ociosos (inativos) de acordo com os tipos de equipamentos e seu consumo em atividade. As tabelas 4.1, 4.2 e 4.3 [ENE2000] apresentam as determinações para computadores e monitores.

TABELA 4.1 - Limites de consumo para computadores

<i>Consumo durante operação normal</i>	<i>Consumo máximo enquanto em baixa energização</i>	<i>Tempo para redução do consumo</i>
$\leq 200W$	$\leq 30W$	15-30 minutos
$> 200W$	$\leq 15\%$ do consumo em operação normal	15-60 minutos

Fonte: [ENE2000]

TABELA 4.2 - Limites de consumo para monitores

<i>Modo de baixo consumo</i>	<i>Consumo máximo em baixa energização</i>	<i>Tempo para redução de consumo</i>
1º modo	$\leq 15W$	15-30 minutos
2º modo	$\leq 8W$	< 70 minutos

Fonte: [ENE2000]

TABELA 4.3 - Limite de consumo integrado

<i>Consumo máximo em baixa energização</i>	<i>Tempo para redução de consumo</i>
$\leq 60W$	15-30 minutos

Fonte: [ENE2000]

Os equipamentos que seguem essas determinações ganham o direito de exibir o logotipo do programa. Para incentivar sua adoção, o governo dos Estados Unidos usa o seu próprio poder de compra: determinou que todos os equipamentos comprados pelo governo estejam de acordo com o programa.

Equipamentos compatíveis com o programa *Energy Star* são interessantes para este trabalho por possuírem mecanismos de redução de consumo quando ociosos. Alguns equipamentos permitem que estes mecanismos sejam acionados por *software*, o que possibilita o seu controle através do sistema proposto neste trabalho.

No entanto, alguns equipamentos que possuem estados de consumo reduzidos apresentam barreiras à utilização em rede devido à perda das conexões quando o equipamento entra em estado de consumo reduzido. Esse é um dos motivos pelos quais o controle de consumo é freqüentemente desativado de ambientes de rede [NOR97,MCQ97]. Novas tecnologias, como o ACPI, tentam superar esse obstáculo.

4.2 Mecanismos de Controle de Consumo

Nos equipamentos com gerência de energia, o controle de consumo é realizado através do desligamento de subsistemas ociosos. Determinados equipamentos, como computadores, exigem salvamento de contexto antes de atuar no *hardware*, de maneira a salvar os dados do usuário.

O controle de consumo compreende quatro situações distintas [NOR97]:

- detecção de ociosidade: o BIOS¹¹ ou o sistema operacional verifica atividade do usuário (teclado e *mouse*), dos processos e de dispositivos internos (*modem*, placa de rede);
- temporização: o sistema mantém um temporizador para cada dispositivo monitorado; esse temporizador estabelece limites de tempo sem uso, para que o dispositivo possa ser considerado ocioso;
- atuação: pode-se atuar no equipamento inteiro, quando não há nenhuma atividade ou nos dispositivos de maneira isolada, colocando-os em estados de baixo consumo. Podem existir diferentes estados de baixo consumo que se diferenciam pela quantidade de energia economizada e pelo tempo necessário para retorno ao estado de atividade normal;
- retorno à atividade normal: o sistema deve monitorar dispositivos para verificar a ocorrência de atividade e retornar ao estado de atividade normal.

11 BIOS – *Basic Input Output System* – *firmware* que armazena a configuração do computador.

Para o controle de consumo, é necessário que todas as camadas do sistema computacional - *hardware*, BIOS, sistema operacional e aplicações - sejam desenvolvidas com este objetivo.

É necessário também que o *hardware* possa ter sua atividade monitorada e controlada pelo *software*. Para um controle adequado, o *software* deve ser capaz de monitorar a atividade e de controlar o estado de funcionamento do equipamento e de cada dispositivo interno separadamente. Não apenas a característica ligado/desligado é empregada para atuar sobre o consumo de energia; alguns dispositivos de *hardware* podem ter seu consumo reduzido pela redução da velocidade de operação, como o processador, por exemplo.

O BIOS deve estabelecer a interface entre o *hardware* e o *software*, oferecendo serviços para as camadas superiores. Em alguns sistemas computacionais, o BIOS coordena a gerência de energia do equipamento.

O sistema operacional ajusta a estratégia de gerência de energia em função das necessidades das aplicações. Também deve informar às aplicações e ao usuário sobre as medidas de redução de consumo adotadas.

As aplicações devem informar ao sistema operacional quais recursos são necessários para sua execução. Também devem racionalizar o acesso a recursos que possam ser gerenciados, como o disco rígido, por exemplo.

Por necessitar de envolvimento de todas as camadas, os mecanismos de gerência de energia são dependentes de plataforma de *hardware* e do sistema operacional. Assim, além de cada plataforma possuir seu próprio mecanismo de controle de consumo, ainda fabricantes de plataformas similares podem utilizar mecanismos diferentes. Por consequência e devido à diversidade de fabricantes, computadores com arquitetura Intel passaram a necessitar de padrões de gerência como forma de atingir limitações nesta “criatividade”.

Os mecanismos usados na gerência não agem necessariamente sobre toda a máquina. Muitas vezes eles exploram principalmente o desligamento de monitores, de disco rígido e do próprio processador (este normalmente recebe atuação apenas para reduzir a frequência de relógio). Suas principais características são expostas a seguir.

O monitor consome mais da metade da energia do sistema computacional inteiro. Os mecanismos de controle de consumo permitem redução de consumo em até 95% com tempo de retorno de 5 segundos [NOR97]. A gerência do monitor não interfere no funcionamento do equipamento (desde que não seja utilizado interativamente). Por esses motivos, o desligamento do monitor é recomendado quando o equipamento possui fonte limitada de energia mas precisa continuar operando de forma não interativa.

O disco rígido pode ser colocado em estado de baixo consumo após períodos sem utilização. Quando existe necessidade de buscar dados do disco, este é reativado automaticamente pelo sistema. A desvantagem de desativar o disco rígido é o tempo de retorno ao estado normal, que pode demorar vários segundos, dependendo do modelo de disco rígido.

O processador pode operar com frequência de relógio reduzida para diminuir o consumo e a dissipação de calor, sem deixar de funcionar. O sinal de relógio pode até ser cortado para prover maior redução no consumo; o retorno ao estado normal é rápido.

Equipamentos que não possuem mecanismos de controle de consumo podem ser gerenciados utilizando-se equipamentos externos de controle. Estes equipamentos são atuadores acionados por sinais elétricos. Dessa maneira, pode ser gerado um sinal elétrico que ligue e desligue estes equipamentos.

Com o desenvolvimento tecnológico, outros dispositivos poderão ser controlados no futuro, como placas de som, placas de rede, placas *fax/modem*. Da mesma forma, dentro de algum tempo, o computador deverá estar apto a controlar o consumo de dispositivos externos (equipamentos) como impressoras, por exemplo.

Nas próximas seções deste capítulo, serão apresentados mecanismos de controle encontrados em alguns equipamentos e padrões de controle usados em arquiteturas Intel.

4.3 Computadores Apple Macintosh

A gerência de energia em computadores Macintosh, da Apple, é bastante similar a encontrada em sistemas da plataforma x86 da Intel.

Alguns modelos de PowerMacs utilizam um *software* de gerência de energia que permite ao usuário estabelecer os tempos de ociosidade permitidos para a CPU e o monitor, antes de prover seu desligamento. Esse *software* é fornecido junto com o sistema, mas precisa ser instalado e habilitado pelo usuário para funcionar.

Alguns modelos Quadra e PowerMac desligam-se completamente como estratégia de gerência, ao invés de entrar em estados graduais de baixo consumo. Esta medida força a reinicialização da máquina para que ela possa retornar ao estado ativo, o que ocasiona um grande tempo de retardo, causando também a perda das conexões de rede. Por isso, a gerência de energia raramente é realizada nestas máquinas pois freqüentemente é desabilitada por parte dos usuários. Computadores Apple Macintosh dos modelos Macintosh II, Quadra, Portable e Powerbook, que possuem sistema operacional Mac OS 8 ou 9, podem ser desligados por *software* [APP99]. O sistema operacional possui um gerente de desligamento (*shutdown manager*). É possível executar os procedimentos de desligamento já existentes ou criar procedimentos específicos para dispositivos.

É possível desenvolver agentes de *software* para este sistema que acionem os mecanismos de controle quando orientados pelo gerente. Este agente pode desligar o computador ou o monitor quando comandado pelo gerente.

4.4 Estações Sun Microsystems

Estações Sun *desktop* vendidas após janeiro de 1994 comportam-se de acordo com as exigências do programa *Energy Star*, mas sistemas maiores (servidores) não possuem essa característica.

A gerência de energia nas estações é semelhante a dos PCs, permitindo que monitores e outros dispositivos (como discos rígidos) sejam desligados de forma independente, dependendo de sua atividade.

Quando disponível, a gerência de energia em monitores Sun é realizada através de sinais DPMS¹² (*Display Power Management Signaling*) como em PCs. Alguns dispositivos externos (por exemplo, *scanners* e discos) são desligados após sinalização do processador.

Apesar das semelhanças com os PCs, a gerência do sistema apresenta características próprias. Existe apenas um modo de baixa energização. Esse estado é chamado de *suspend* pelo fabricante e equivale ao modo *hibernate* (seção 5.6.1) nos PCs. A estação pode ser colocada nesse modo se estiver ociosa, ou se um temporizador for programado para esse fim.

Quando a estação entra no modo de baixa energização, todos os processos são suspensos, os dispositivos externos são informados de que devem ser desligados, o contexto é salvo em disco e o sistema é desligado. Nesse estado, a estação ainda consome uma quantidade mínima de energia mas pode ser completamente desligada sem prejuízo do contexto salvo.

O sistema pode retornar ao estado ativo através do acionamento da tecla “*power*” ou através da ação de um temporizador previamente programado (característica disponível apenas em alguns modelos). A latência de retorno ao estado ativo é geralmente inferior a um minuto.

Assim como nas máquinas da Apple (4.3), também neste tipo de equipamento está prevista a possibilidade de um agente que atue nos dispositivos, principalmente no monitor, e no equipamento inteiro.

4.5 O Padrão Advanced Power Management

Este padrão, cuja abreviatura é APM, é utilizado na arquitetura PC; foi desenvolvido para ser utilizado em *notebooks* mas posteriormente foi incorporado em computadores *desktop*. Consiste na especificação da interface entre camadas de *software* (BIOS, sistema operacional, aplicações) para gerência de consumo, sendo o BIOS o principal responsável pela gerência [INT96].

¹² Este sistema permite sinalizar para que o monitor entre em estado de baixo consumo. A sinalização é enviada através do cabo de dados do monitor e consiste na codificação dos sinais de sincronismo vertical e horizontal.

O APM controla a utilização de energia em função do nível de atividade do sistema. Assim que a atividade diminui, a energia de dispositivos que não estejam sendo utilizados é cortada até que o sistema seja levado a um estado de consumo mínimo.

O APM permite duas formas de realização da gerência de energia:

- o BIOS gerencia a energia do sistema com base na utilização dos dispositivos;
- um *driver* do sistema operacional assume a responsabilidade pela estratégia de gerência de energia, trabalhando em cooperação com o BIOS.

Para gerência de energia, sistemas construídos com o objetivo de atender ao padrão APM possuem três camadas de *software* distintas: BIOS, *driver* do sistema operacional e aplicações, de acordo com a figura 4.1, as quais são explicadas na seqüência.

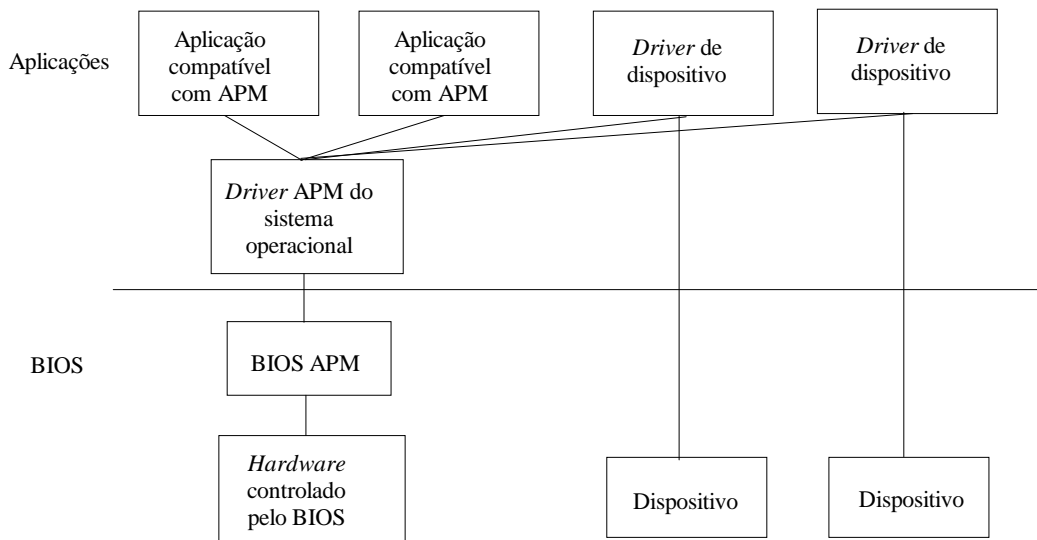


FIGURA 4.1 - Componentes de um sistema APM

O BIOS controla a gerência de energia dos componentes de *hardware*, sendo projetado de forma específica para cada plataforma de forma estritamente vinculada à máquina. O BIOS pode realizar gerência de energia por iniciativa própria, quando o SO não possui *driver* compatível com APM.

Deve existir apenas um *driver* APM do sistema operacional e este deve fornecer uma interface de comunicação comum para todas as aplicações. A especificação dessa interface é realizada pelo fabricante do *driver*. São conhecidos *drivers* para MSDOS, Windows e Linux. Para MSDOS e Windows, a própria Microsoft fornece os *drivers* junto com o sistema operacional.

O *driver* do sistema operacional possui três funções principais:

- realizar a interface entre aplicações e BIOS;
- coordenar a estratégia de gerência em ambientes multitarefa;
- identificar oportunidades de gerência não aparentes ao BIOS e às aplicações.

Após estabelecida a conexão, o *driver* do sistema operacional realiza consultas ao BIOS a intervalos regulares, sendo recomendada uma consulta por segundo. Caso o BIOS informe que o sistema deve entrar nos modos *standby* ou *suspend*, o *driver* do sistema operacional deve repassar a informação às aplicações e aos *drivers* de dispositivo para que estes executem os procedimentos apropriados de salvamento de contexto; somente após o BIOS é notificado de que pode continuar a operação.

As aplicações podem contribuir para a gerência informando ao SO sobre os recursos necessários para o processamento. Nessa camada também encontram-se os *drivers* de dispositivo que fornecem gerência de energia para dispositivos que não sejam gerenciados pelo BIOS como adaptadores de vídeo, por exemplo. A especificação não impõe restrições a aplicações e *drivers* de dispositivos que não utilizam o padrão APM.

Aplicações que participam da gerência de energia são registradas pelo *driver* do sistema operacional e informadas sobre os eventos de gerência. Segundo a especificação, a documentação do *driver* do sistema operacional deve fornecer uma descrição da maneira pela qual o registro deve acontecer.

Os *drivers* de dispositivo são responsáveis pelo salvamento e recuperação do contexto dos dispositivos por eles gerenciados.

4.5.1 Estados de Energização do Sistema

O APM define cinco estados de energização para o sistema: *full on*, *enabled*, *standby*, *suspend* e *off*. Estes estados são definidos em termos gerais, permitindo variações na implementação, e são apresentados a seguir:

- quando a máquina está no modo *full on*, a gerência de energia não é realizada. O sistema processa normalmente e os dispositivos permanecem energizados. Os únicos estados possíveis a partir deste são o estado *enabled* e o estado *off*;
- no estado *enabled*, a gerência de energia está habilitada. O sistema processa normalmente mas o sinal de relógio do processador é reduzido ou cortado, quando possível. Os dispositivos também podem ser gerenciados nesse estado;
- o sistema entra no modo *standby*, após curto período sem atividade. Neste estado, o processamento pode ser interrompido e dispositivos não utilizados são colocados em estados de consumo mínimo de energia. O tempo de retorno ao estado *enabled* é pequeno;
- se o sistema for mantido ocioso, é colocado no estado *suspend*. Nesse estado, o sistema pára de processar. O tempo de retorno ao estado *enabled* é grande. O APM permite a implementação do estado *hibernate* como uma implementação especial do estado *suspend*, mas a existência desse estado não é obrigatória;
- no estado *off*, o fornecimento de energia da máquina é interrompido. O sistema deve ser reinicializado para retornar ao estado ativo.

O estado *hibernate* permite o salvamento do contexto do sistema (conteúdo da memória) em disco e o posterior desligamento do *hardware*. Ao ser ligado novamente, o sistema continua processando a partir do ponto em que parou. Apesar de não obrigatório, este modo foi bastante implementado em *notebooks*.

Quando o sistema passa ao estado *suspend*, todos os dispositivos podem ter a energia cortada. Por isso o sistema operacional requisita antecipadamente que todos os *drivers* preparem os dispositivos para o desligamento. Por esse motivo, o sistema operacional pode retardar ou mesmo rejeitar um pedido de suspensão do BIOS ou do usuário.

4.5.2 Estados de Energização de Dispositivos

Os dispositivos podem ser gerenciados pelo BIOS ou pelo sistema operacional. Através da execução de uma rotina, o sistema operacional pode solicitar ao BIOS para deixar de gerenciar um determinado dispositivo. Nesse caso, um *driver* de dispositivo específico pode gerenciar esse dispositivo, cabendo a esse *driver* o salvamento e a recuperação de contexto do dispositivo quando o estado de energização mudar.

Os dispositivos podem estar em um dos quatro estados abaixo:

- *device on*: o dispositivo está ativo e não é gerenciado;
- *device power managed*: o dispositivo está ativo mas pode ter seu desempenho reduzido ou alguma funcionalidade desativada em função do uso. Podem existir diferentes níveis de gerência para este estado;
- *device low power*: o dispositivo não está ativo mas continua sendo energizado, de forma que seu contexto é mantido. A especificação também permite a existência de diferentes modos de baixa energização;
- *device off*: o dispositivo não está ativo e a energia é cortada. O contexto é perdido.

A especificação permite que dispositivos possuam controle próprio de energia, mas, não os leva em consideração na gerência.

4.5.3 Estados de Energização do Processador

O processador¹³ é gerenciado de maneira particular. Normalmente é o último componente a ser desligado e o primeiro a ser ligado.

Quando o sistema está nos estados *enabled* ou *standby*, o sinal de relógio do processador é reduzido ou desligado e ligado conforme necessário, mas outros circuitos periféricos (como memória *cache*, barramento e temporizadores) não são alterados. No modo *suspend*, o sinal de relógio do processador é cortado e os outros circuitos são colocados em estado de baixa energia.

¹³ Alguns processadores são projetados para operar em estados de consumo reduzido, quando possível. Este trabalho não aborda o controle de consumo nesse nível, apenas a interface relevante do BIOS.

O *driver* APM do SO pode notificar o BIOS sobre a utilização do processador, mas é sempre o BIOS que realiza a gerência do processador.

São definidos três estados possíveis para o processador:

- *full on*: nesse estado, o processador possui desempenho máximo e alto consumo;
- *slow clock*: o sinal de relógio é reduzido. O desempenho cai e o consumo diminui;
- *stop*: apenas interrupções de *hardware* podem reativar o sinal de relógio.

O tempo de retorno ao estado *full on*, a partir de qualquer um dos outros dois estados não é perceptível ao usuário.

4.5.4 Transições de Estados de Energização

Em um sistema adequado ao padrão APM, o BIOS monitora a atividade do sistema e o leva a estados sucessivos de consumo reduzido. O *driver* do sistema operacional apenas realiza chamadas de funções para que o BIOS realize as transições de estados. A figura 4.2 ilustra a maneira pela qual as transições de estados ocorrem.

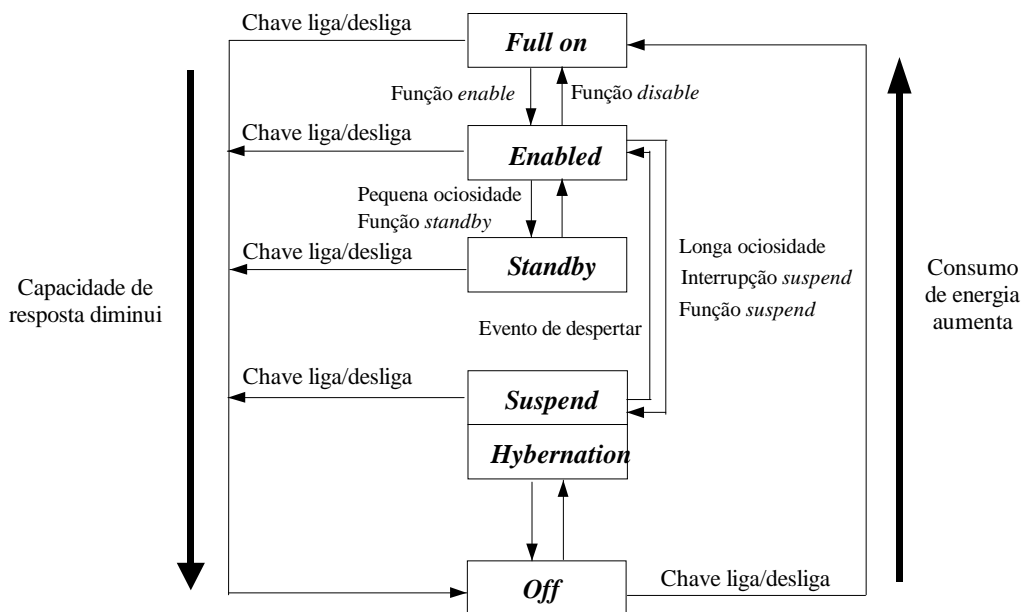


FIGURA 4.2 - Transições de estado em um computador com APM

A utilização do APM para gerência de energia consegue a redução do consumo de energia para 25% do consumo total, quando o computador está no modo de energização mínimo [NOR97].

4.5.5 Interface de Funções

A especificação APM determina que o BIOS proporcione um conjunto de funções através da interrupção 15. Dessa maneira, é possível controlar esses computadores independentemente do sistema operacional utilizado.

A maioria das funções exige acesso exclusivo, o que significa que apenas um *software* pode utilizá-las de cada vez. Como deve existir no sistema somente um *driver* do sistema operacional, o agente proposto por este trabalho deve utilizar os serviços do *driver* do sistema operacional, quando possível, ou suplantá-lo caso seja necessário.

4.5.6 Considerações sobre o APM

O projeto inicial do PC (no início da década de 1980) não previa gerência de energia, visto que não existem linhas independentes de fornecimento de energia para os dispositivos. A especificação APM, por sua vez, descreve apenas aspectos de *software*, deixando a implementação do *hardware* bastante livre. Por esse motivo, o suporte de *hardware* necessário à gerência de energia não foi desenvolvido comercialmente.

O BIOS administra o gerência de energia em alguns dispositivos e não incorpora ao esquema de gerência os acrescentados posteriormente. Com a adição de novos dispositivos, cabe aos *drivers* destes dispositivos a tarefa de gerência de energia (figura 4.1).

A especificação não determina que o sistema possa ser desligado por *software* e não define eventos de despertar. Assim, embora tenha sido importante como primeira estratégia de gerência de energia padronizada, o APM impõe restrições à gerência completa da máquina.

4.6 O Projeto OnNow

O projeto *OnNow Design Initiative* da Microsoft [MIC99] é composto de uma série de especificações que propõem mudanças na arquitetura de *hardware* e *software* para que os PCs estejam prontos para utilização instantaneamente quando ligados, como eletrodomésticos. Para isto, o computador é colocado em um estado de consumo reduzido, que parece desligado para o usuário. Estes estados de baixo consumo são chamados de estados de espera, ou *sleeping states*, em inglês.

Esse projeto introduziu um novo conceito para controle de consumo em computadores, chamado OSPM (*Operating System Power Management*), que leva a gerência de energia do BIOS para o sistema operacional. Apesar da gerência de energia necessitar de cooperação entre todas as camadas, a tomada de decisão deve ficar a cargo do SO [LOR97], pois as camadas inferiores (*hardware* e BIOS) não possuem informações suficientes sobre a carga do sistema e as camadas superiores devem abstrair-se de questões de baixo nível, como acionar ou desativar um dispositivo, por exemplo.

Quando em estado de espera, o computador deve ser capaz de responder prontamente a eventos como atuação sobre a chave liga/desliga e chegada de pacotes da rede. Algumas características do projeto OnNow são:

- elimina retardos de ligação e desligamento, pois ao invés de realmente ligar e desligar, o computador entra e sai de estados de espera;
- permite a execução automática de tarefas, quando o computador está em estado de espera, ou seja, o usuário pode programar processos para serem executados mesmo quando o sistema estiver nesse estado. Nesse caso, o computador é despertado no momento desejado e colocado novamente em estado de espera, quando o processamento acabar;
- permite gerência de energia por dispositivo, ou seja, dispositivos que não estejam sendo utilizados são desligados.

De acordo com o previsto no OnNow, existem quatro estados possíveis para o computador:

- *working* (ativo): o sistema apresenta toda a sua funcionalidade; dispositivos não utilizados são colocados em estado de consumo reduzido;
- *sleeping* (espera): o sistema parece desligado; o consumo é reduzido para um dos níveis permitidos;
- *soft off* (quase desligado): parece desligado; consumo muito baixo; o contexto pode ser recuperado se for salvo em meio não volátil;
- *mechanical off* (desligado): sem consumo; o sistema retorna ao estado ativo após reinicialização do sistema operacional.

4.6.1 Classes de Dispositivos

Para uma gerência de energia eficiente, é necessário que o sistema conheça as características individuais de cada dispositivo. Por exemplo, uma placa de som possui características e funcionalidades diferentes de uma placa de vídeo, quando ambas operam no mesmo estado de energização.

Cada classe de dispositivos possui seu próprio conjunto de políticas de gerência de energia, que define o comportamento em cada estado de energização. Esse conjunto é padronizado de maneira que os fabricantes possam utilizar estas políticas em seus produtos com garantia sobre a interoperabilidade dos dispositivos.

Para possibilitar a padronização, os diferentes dispositivos são agregados em classes, às quais é associada uma política de gerência que deve abordar os seguintes aspectos [MIC98a]:

- comportamento: cada classe deve possuir uma definição para cada estado de energização, incluindo níveis de consumo de energia, tempos de latência de resposta a comandos e tempos de latência de mudanças de estado;
- conjunto mínimo de características a serem implementadas: pode ser suficiente implementar apenas alguns estados de energização para

determinada classe ou existirem exigências específicas relacionadas a determinados eventos. Por exemplo, é uma exigência para a classe de *modems* a capacidade de despertar o computador a partir de determinado estado;

- características funcionais: deve ser definido o conjunto de funcionalidades que estará disponível em cada estado de energização para cada classe. Como exemplos, tem-se que, em determinado estado, o adaptador de rede pode receber pacotes mas não pode transmitir, ou a placa de som está completamente funcional, mas os amplificadores estão desligados.
- evento de despertar: é tratado de maneira especial, pois o gerente de energia deve conhecer previamente quais dispositivos podem despertar a máquina. A política deve determinar se a classe pode ou não despertar o computador, quais eventos devem ser monitorados e qual deve ser o tempo de latência da operação, entre outros.

Os sistemas podem implementar apenas as três primeiras, sendo opcional a função de despertar.

As políticas de gerência para cada classe de dispositivos são determinadas pela estratégia *OnNow* e estão disponíveis para obtenção pela Internet [MIC98b]. As seguintes especificações estão disponíveis:

- *Default Class Specification*: é direcionada a todas as classes, mesmo as que possuem especificação própria;
- *Audio Specification*: define políticas de gerência de dispositivos de áudio;
- *Communications Specification*: define políticas para portas de comunicação e *modems*;
- *Display Specification*: é aplicada em monitores e controladores de vídeo;
- *Input Specification*: desenvolvida para dispositivos de entrada de dados, como teclado, *mouse*, *joystick* e outros;
- *Network Specification*: descreve as políticas de gerência para adaptadores de redes *token ring* e *ethernet*;
- *PC Card Specification*: descreve políticas para dispositivos *PC Card*;
- *Storage Specification*: especifica políticas de gerência para discos flexíveis, *drives* de CDROM ATAPI e SCSI e discos rígidos ATA e IDE.

4.6.2 Funções de Gerência de Energia

Existem basicamente quatro funções padronizadas que devem ser implementadas para cada classe [MIC98b]. Essas funções devem ser implementadas pelos *drivers* dos dispositivos e são executadas pelo sistema operacional:

- obter habilidades (*get capabilities operation*): o resultado desta operação informa ao SO as habilidades e características de um determinado dispositivo. É realizada na inicialização da máquina, após a identificação de dispositivos

que o sistema operacional executa. Como essas informações incluem os estados de energização e as habilidades de despertar implementadas, o sistema operacional as utiliza para ajudar a determinar as políticas de gerência de energia a serem adotadas.

- ativar estado de energização (*set power state operation*): esta operação leva o dispositivo a determinado estado de energização e habilita características de gerência com base na política global preestabelecida e nas habilidades específicas do dispositivo;
- obter estado de energização (*get power status operation*): esta operação fornece informações para o sistema operacional sobre o estado atual de energização do dispositivo;
- despertar (*wakeup operation*): esta operação possibilita ao dispositivo despertar o sistema de um estado de espera, exemplo, uma placa de rede ao detectar um pacote de interesse.

4.6.3 O Padrão *Advanced Configuration and Power Interface*

O padrão *Advanced Configuration and Power Interface* (abreviado por ACPI) [INT96a] também descreve a interface entre os componentes de *hardware* e *software* para gerência de PCs. A especificação ACPI é parte integrante da estratégia *OnNow* e foi proposta para substituir o APM como interface de gerência de energia em PCs. Cobre as seguintes áreas:

- gerência de energia do sistema: define mecanismos para que o SO possa realizar transições de estados de energização do sistema;
- gerência de energia dos dispositivos: define tabelas que descrevem as características dos dispositivos e os mecanismos de controle para a gerência destes;
- gerência de energia do processador: de acordo com o grau de inatividade o sistema operacional pode colocar o processador em estados de baixo consumo;
- *plug and play*: define mecanismos para a identificação e configuração dos dispositivos;
- eventos de sistema: define mecanismos para que o sistema possa lidar com eventos térmicos, de gerência de energia, *docking*¹⁴, inserção de dispositivos¹⁵, entre outros;
- gerência de baterias: confere ao SO o controle dos níveis baixo ou crítico de carga de baterias de equipamentos portáteis. Também fica responsável por calcular a capacidade restante da bateria. No APM, o BIOS realizava essa tarefa;
- gerência térmica: o SO passa a possuir mecanismos para controlar a temperatura de operação do sistema;

¹⁴ Acoplamento de *notebooks* a gabinetes *desktop*.

¹⁵ O ACPI permite que o sistema reconheça novos dispositivos sem necessidade de recarga do SO.

- controladores de sistema: define uma interface para que o SO possa se comunicar com controladores construídos no *hardware* (*embedded controllers*);
- controle de barramento: semelhante ao item anterior, o SO pode se comunicar com controladores de barramento do sistema.

Esta seção apresenta apenas a descrição dos mecanismos de gerência de energia, presentes na especificação, que estão listados nos três primeiros itens acima. A gerência de baterias não é abordada por estar relacionada ao consumo local do equipamento. Nesse caso quem deve gerenciar o consumo é o próprio sistema operacional e não o gerente de consumo da rede.

4.6.4 Estrutura

A figura 4.3 [INT96a] mostra os componentes de um sistema ACPI e ilustra as relações entre eles e o restante do sistema (sistema operacional, aplicações e *hardware*).

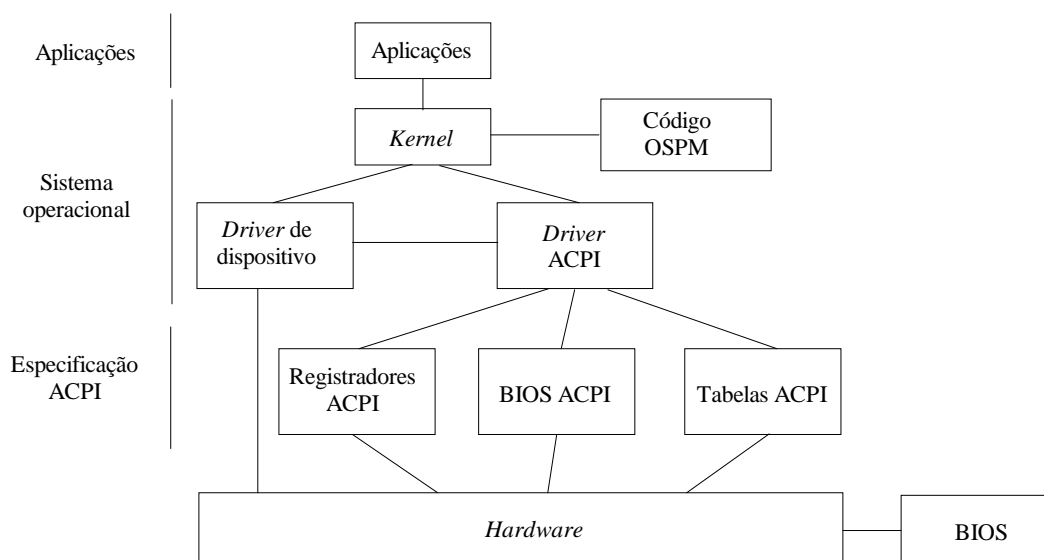


FIGURA 4.3 - Estrutura do ACPI

Além dos componentes normalmente encontrados em um equipamento de computação, a figura 4.3 apresenta componentes que são específicos a sistemas compatíveis com o ACPI, que são:

- as tabelas contêm descrições do *hardware*; podem conter também seqüências de operações necessárias para a execução de funções no *hardware*. O SO possui um interpretador para essas instruções inserido no driver ACPI do sistema operacional. As tabelas são integrantes do BIOS, apesar da separação aparente das estruturas na figura 4.3;
- os registradores realizam o controle do *hardware* descrito nas tabelas;
- o BIOS é a parte do *firmware* que implementa a especificação. O código de instruções do BIOS faz a inicialização da máquina e implementa operações de gerência do *hardware*.

O módulo chamado de código OSPM (*Operating System Power Management*) fornece o *software* necessário para que o sistema operacional possa gerenciar o consumo do equipamento.

A figura 4.3 também mostra que é permitida a existência de dispositivos com *drivers* específicos que não são compatíveis com o ACPI.

4.6.5 Estados de Energização

A figura 4.4 [INT96a] contém uma representação gráfica dos estados de energização definidos pelo padrão e suas interrelações.

Os sistemas não precisam implementar todos os estados definidos, mas devem implementar pelo menos um estado de espera (S1-S4).

Durante o período de utilização, o computador é colocado alternadamente nos estados ativo (*working*) e de espera (*sleeping*). No estado ativo, G0 na figura 4.4, o computador possui toda a funcionalidade disponível. Dispositivos e processadores que não estejam sendo utilizados podem ser colocados em baixa energização (estados Dx e Cx, respectivamente, sendo $x = 0, 1, 2$ ou 3). Os dispositivos que estejam em baixa energização podem retornar ao estado ativo em um período curto de tempo (o valor deste “período curto” depende do dispositivo). No estado ativo, o computador pode operar com diferentes velocidades de processamento, de modo a refletir o ajuste desejado entre velocidade, consumo, aquecimento e emissão de ruídos.

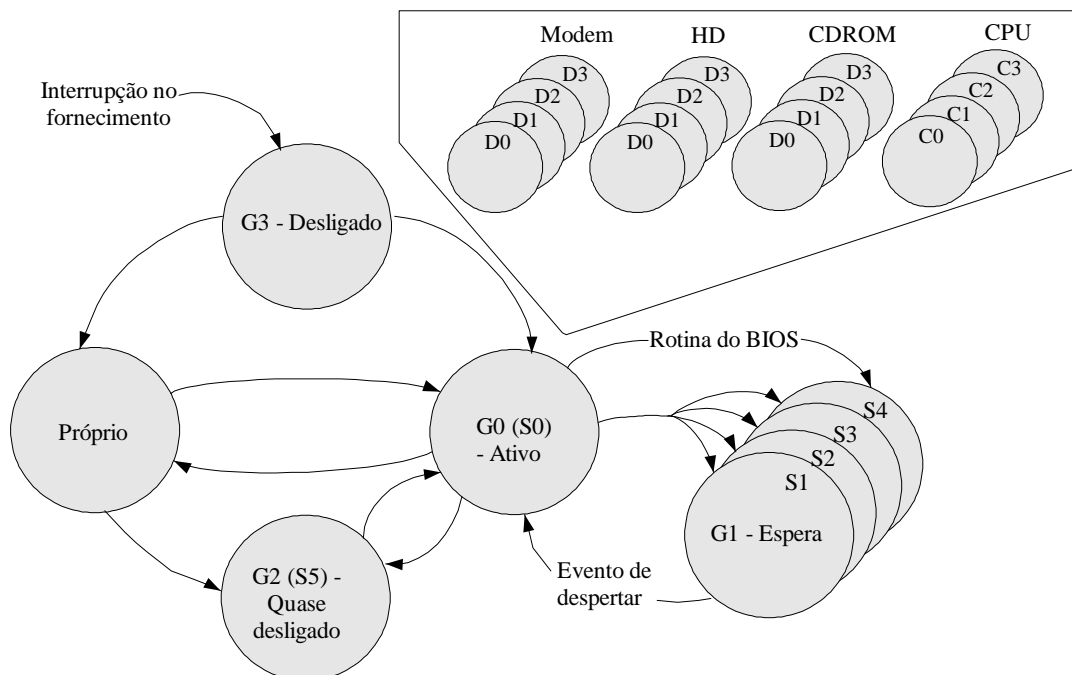


FIGURA 4.4 - Estados de energização no ACPI

Quando o computador está ocioso ou quando o usuário pressiona a chave liga/desliga, o SO conduz o computador a um estado de espera ou ao estado de *soft off*. Os estados de espera são designados genericamente de G1 e cada estado específico é

denominado Sx, conforme a figura 4.4. *Soft off* (G2 ou S5) é o estado de menor consumo em que o SO pode colocar o equipamento, mas não é um estado de espera porque nenhuma informação de contexto é salva. Esse estado possui as seguintes características:

- nenhum programa de usuário ou de sistema é executado;
- quase todos os circuitos são desligados, havendo perda de contexto e tornando necessária a reinicialização do computador para que este retorne ao estado ativo.

As exceções, ou seja, os únicos circuitos energizados, são os circuitos responsáveis por despertar o computador na presença de eventos. O diferencial desse estado para o desligado é que o sistema pode ser despertado a partir de eventos como uma chamada telefônica, o acionamento de uma tecla ou a detecção de um pacote da rede.

O SO deve levar o computador a estados de “espera superficial” (S1 a S3) se o usuário desejar que a máquina retorne ao estado ativo na ocorrência de qualquer evento de despertar ou se o tempo de retorno precisar ser curto. O SO pode colocar a máquina em estado de “espera profunda” se uma latência de minutos for permitida. O significado dos termos “espera superficial” e “espera profunda” depende das configurações feitas pelo usuário.

Os estados S1 e S2 são estados de espera superficial em que o consumo é reduzido de maneira branda.

O estado S3, chamado de *Suspend to RAM*, permite redução em mais de 95% do consumo total [INT97] e realiza o salvamento do contexto em memória RAM.

O estado S4, chamado de *Non Volatile Sleep* ou NVS, é um estado de espera semelhante ao *hibernate* mencionado na seção 4.5.1. O contexto do sistema é salvo em um arquivo e o *hardware* é levado a um nível de desligamento semelhante ao do estado S5. O contexto será recuperado quando o computador “despertar”, caso as três condições seguintes sejam satisfeitas:

- seja encontrado um arquivo NVS válido;
- não tenha havido alteração de aspectos de configuração, como tamanho da memória e organização do disco rígido;
- a operação de recuperação de contexto não seja interrompida pelo usuário.

Do ponto de vista do usuário, o sistema comporta-se como se estivesse retornando do estado G1, embora mais lentamente.

Na ausência de *driver* ACPI no sistema operacional, a especificação permite que o computador utilize mecanismos próprios de controle de consumo. Quando um *driver* ACPI é carregado, o computador é levado ao estado G0. Computadores que não possuem mecanismos de controle de consumo passam diretamente de G3 para G0.

Para finalizar, cabe observar que a intenção do ACPI é que os usuários não desliguem mais os computadores, apenas colocando-os em estados de baixo consumo.

Assim, ao invés de cortar o fornecimento de energia ao sistema, a chave liga/desliga envia um sinal que o SO interpreta como uma requisição para que o sistema seja colocado em um estado de espera profunda ou de *soft off*, de acordo com preferências pré-definidas pelo usuário. Os sistemas devem possuir uma segunda chave que realmente realize o corte de energia, para os casos em que esta condição seja efetivamente necessária. A legislação de alguns países torna obrigatória a existência dessa segunda chave por medida de segurança (desligamento emergencial).

4.6.6 Considerações sobre o ACPI

O ACPI é bastante abrangente pois não aborda apenas aspectos de gerência de energia. A detecção de eventos do sistema e a detecção de inserção e remoção de dispositivos permite que o SO reconheça dispositivos adicionados, sem necessidade de reinicialização do sistema.

O ACPI descreve também a interface de *hardware*, de modo a padronizar a implementação dos dispositivos, e permite que o computador seja desligado por *software* e despertado por eventos.

Visando adequar os computadores à especificação, a implementação do *hardware* deve ser modificada para fornecer linhas de alimentação individualizadas para os dispositivos. Essa característica foi incorporada ao padrão ATX [INT98], que define a disposição dos componentes da placa mãe de PCs. Dentre as determinações específicas de projeto [INT98a], pode-se citar a exigência de que a fonte de alimentação seja ligada e desligada por *software*; além de ser prevista como necessária a alimentação independente dos diversos subsistemas.

4.7 Equipamentos sem Suporte à Gerência de Energia

Equipamentos que não possuam mecanismos próprios de gerência necessitam de *hardware* adicional, chamado de chave eletrônica no decorrer do texto, para que a gerência seja possível. Chaves eletrônicas existem no mercado [BAY99, BUS99, DAT99, IND99, SER99] ou podem ser facilmente implementadas; o requisito necessário à gerência é que tenham interface de comunicação.

As chaves eletrônicas são colocadas entre o equipamento que se deseja gerenciar e o ponto de fornecimento de energia. A figura 4.5 apresenta um exemplo de utilização para que o computador controle a energização de um *hub*. Deve ser desenvolvido um agente *proxy* (seção 3.2) para permitir ao gerente controlar esta chave. Com chaves eletrônicas é possível apenas ligar ou desligar equipamentos, sendo o salvamento de contexto realizado pelo próprio equipamento controlado.

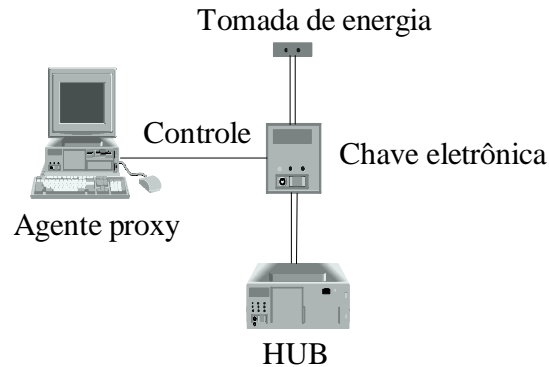


FIGURA 4.5 - Ligação da chave eletrônica

4.8 Mecanismos dos Sistemas Operacionais

O sistema de gerência proposto utiliza os recursos disponíveis nos sistemas operacionais para a gerência de energia. Alguns sistemas operacionais possuem *Application Program Interfaces*, APIs, de funções que as aplicações podem executar. Em outros casos, é necessário escrever um *driver* para controle de consumo.

Os testes realizados demonstraram que os *drivers* APM funcionam perfeitamente em computadores com ACPI, deduzindo-se que certamente os fabricantes se preocuparam com questões de compatibilidade. Estes equipamentos inclusive disponibilizam os serviços da interrupção 15. Equipamentos compatíveis com ACPI com *driver* APM não possuem certas funcionalidades, como a ligação por detecção de atividade, por exemplo, mas podem ser desligados por *software*.

4.8.1 MSDOS

Computadores que executam o sistema operacional MSDOS¹⁶ podem ter acesso às funções do APM através da interrupção 15 do BIOS; esse sistema operacional possui um *driver* APM chamado "*power.exe*". Este aplicativo normalmente é utilizado em *notebooks*. Se o *driver* "*power.exe*" estiver sendo executado, deve-se utilizar uma prática comum neste tipo de situação, interceptando as chamadas do sistema operacional.

4.8.2 Windows

O Windows 95 possui *driver* para o APM e disponibiliza funções em sua API para a utilização por parte de aplicações. As funções disponibilizadas são as mesmas disponibilizadas no Windows 98 e no Windows 2000 [MIC98, MIC98a, MIC98b], e a seguir citadas:

- *BOOL ExitWindowsEx(UINT uFlags, DWORD dwReserved)*

¹⁶ MSDOS - *Microsoft Disk Operating System* - sistema operacional da Microsoft anterior ao *Windows*.

Esta função permite que a aplicação realize uma série de funções de controle, como desligar o computador, reinicializar o computador e efetuar *logoff* do usuário.

- *BOOL SetThreadExecutionState(EXECUTION_STATE esFlags)*

Essa função permite à aplicação indicar ao sistema que está ocupada para evitar que o sistema seja colocado em estado de espera. Por exemplo, essa função pode ser usada para garantir que o sistema não seja considerado ocioso enquanto exibe um vídeo.

- *BOOL SetSystemPowerState(BOOL .Suspend, BOOL .Force)*

Essa função permite que uma aplicação requisiute que o sistema seja colocado em estado de espera. Os parâmetros indicam se as aplicações devem ser consultadas antes ou se a mudança de estado é imperativa.

BOOL GetSystemPowerStatus(LPSYSTEM_POWER_STATUS lpSystemPowerStatus)

Essa função permite que uma aplicação obtenha informações sobre o estado de energização do sistema.

- *BOOL GetDevicePowerState(IN HANDLE h, IN OUT BOOL *pfOn)*

Essa função permite que uma aplicação obtenha informações sobre o estado de energização de um dispositivo.

Não existe função na API para alterar o estado de energização de dispositivos, sendo necessário atuar em nível mais baixo, desenvolvendo um *driver* do sistema operacional para isso.

Além dessas funções, existe uma mensagem do sistema específica para informar às aplicações as medidas de gerência de energia executadas pelo sistema operacional.

- *WM_POWERBROADCAST*

As aplicações devem tratar essa mensagem para receber notificações de eventos de gerência de energia. Diversos estados de consumo podem ser implementados, conforme listado na tabela 4.4.

TABELA 4.4 - Estados de espera do Windows

<i>Estado de espera</i>	<i>Estado ACPI</i>	<i>Estado APM</i>
<i>PowerSystemWorking</i>	S0	<i>Working</i>
<i>PowerSystemSleeping1</i>	S1	<i>Suspend</i>
<i>PowerSystemSleeping2</i>	S2	<i>Suspend</i>
<i>PowerSystemSleeping3</i>	S3	<i>Suspend</i>
<i>PowerSystemSleeping4</i>	S4	<i>Suspend</i>

Segundo a Microsoft, o Windows NT não possui suporte à gerência de energia. Esse suporte somente foi incorporado à versão substitutiva, o Windows 2000.

Os sistemas operacionais Windows 98 e Windows 2000 são os primeiros sistemas operacionais que possuem *drivers* tanto para APM quanto para ACPI e implementam gerência de energia segundo a recomendação *OnNow*. A API de funções disponibilizada é semelhante a do Windows 95, quando opera com APM. Algumas funcionalidades foram implementadas apenas no Windows 2000, devido ao prazo de lançamento do Windows 98.

4.8.3 Linux

Este sistema operacional possui um *driver* APM e uma API de funções para sua utilização. As funções do APM são utilizadas através do dispositivo¹⁷ `"/dev/apm_bios"` e do arquivo especial `"/proc/apm"`.

Existe uma biblioteca, chamada `"libapm.a"`, que disponibiliza as funções mais utilizadas em sistemas APM.

Segundo informações obtidas ao final de 1999 [ACP99], um *driver* ACPI estava sendo desenvolvido, mas o trabalho encontrava-se em estágio inicial.

¹⁷ O sentido do termo nesse caso é referente ao conceito de dispositivo encontrado nos sistemas operacionais compatíveis com Unix, ou seja, a interface com o APM é implementada como um dispositivo do sistema.

5 Sistema de Gerência Proposto

Este capítulo apresenta o sistema de gerência proposto, que deve permitir o controle de consumo tanto em situações de operação com recursos limitados de energia como em situações de operação normal, objetivando redução no consumo.

As características desejadas e as soluções adotadas no desenvolvimento do sistema foram obtidas a partir do estudo de diversas fontes:

- soluções comerciais apresentadas na seção 2.3 [LIE99, APC99, ENG99];
- plataforma de gerência HP OpenView [HEW97], HEW97a];
- trabalhos relacionados à área de gerência de redes [DAV97];
- projeto de cooperação UFRGS-CP Eletrônica [JAN97], LEB98].

O sistema possui as características a listadas a seguir, explicando-se brevemente as opções adotadas para seu atendimento:

- economia de recursos de rede: escolha do SNMP, o qual foi desenvolvido para atender a esta exigência, comum em sistemas de gerência;
- portabilidade: o SNMP exige apenas que o equipamento possua uma implementação dos protocolos UDP e IP. Atualmente grande número dos equipamentos interligados em rede possuem esses protocolos;
- disponibilidade: esta característica é conseguida através da utilização de técnicas de tolerância a falhas descritas no decorrer do capítulo;
- extensibilidade: o sistema pode operar com qualquer tipo de UPS gerenciável e deve atuar em qualquer tipo de equipamento gerenciável.

O sistema proposto pode ser visualizado na figura 5.1 e possui os seguintes componentes:

- **Interface gráfica:** realiza a interface do sistema com o administrador humano¹⁸. Possui duas funções: permitir a visualização do estado momentâneo de consumo dos equipamentos e a definição das políticas de redução de consumo a serem adotadas.
- **UPS:** fornece energia ao sistema em caso de falha da concessionária. Deve informar o gerente sobre falhas no fornecimento comercial e a situação das baterias.
- **Gerente principal:** executa as medidas necessárias para controlar o consumo da rede. Recebe notificações de eventos da UPS e, de acordo com definições do administrador, atua nos equipamentos da rede através dos agentes. É o componente principal do modelo.

¹⁸ A partir deste ponto referenciado apenas como administrador; o termo gerente é utilizado para identificar o *software* de controle do sistema.

- **Gerente reserva**¹⁹: componente apto a desempenhar atividades de gerência. Deve detectar se o gerente está defeituoso e tomar seu lugar neste caso. Possui o mesmo código do gerente principal.
- **Agentes**: componentes que residem nos equipamentos gerenciados e realizam a interface entre o sistema operacional destes e o gerente. Agentes proxy permitem a gerência de equipamentos que não possam ser gerenciados diretamente.

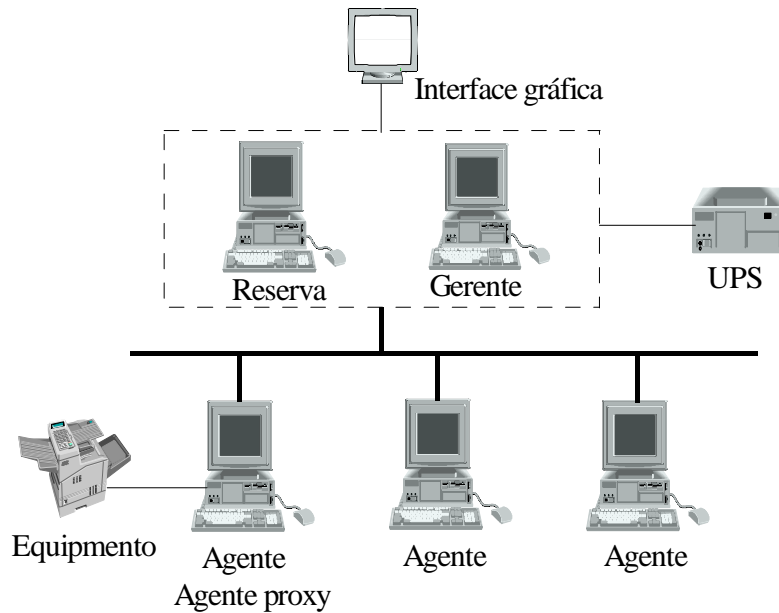


FIGURA 5.1 - Sistema Proposto

5.1 Gerente Principal

O gerente principal armazena as políticas de controle de consumo, determinadas pelo administrador, e o estado atual de consumo das máquinas. Com base nessas informações, e na situação informada pela UPS, decide quais máquinas devem entrar em estado de baixo consumo (ou serem desligadas) e quais podem continuar funcionando.

Possui a estrutura geral ilustrada na figura 5.2. Uma descrição detalhada do gerente é fornecida no capítulo 6. Devido à estrutura modular e à separação entre as estruturas de dados, um gerente pode monitorar diversas UPS que alimentam subredes diferentes.

O módulo de comunicação com a UPS é responsável por detectar os eventos de energia que forem de interesse, principalmente quando a rede passa a operar com energia de baterias. É desejável também que receba notificação sobre a quantidade de energia disponível nas baterias.

¹⁹ A partir deste ponto referenciado apenas por reserva. O termo gerente é utilizado para identificar o gerente principal.

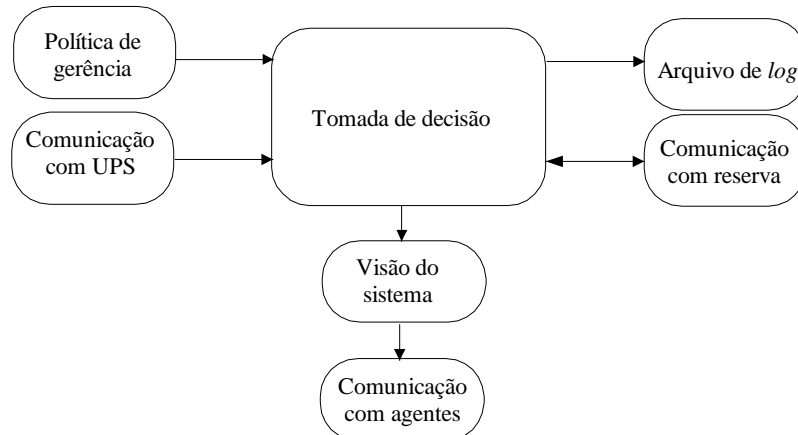


FIGURA 5.2 - Estrutura básica do gerente

O módulo de comunicação com os agentes é responsável por receber notificações sobre transições de estado dos equipamentos e enviar comandos para alterar o estado de consumo destes. Com base nas notificações enviadas pelos agentes, o gerente monta uma visão do sistema, através de uma tabela que indica o estado momentâneo de consumo dos equipamentos.

O arquivo de *log* armazena todas as mensagens trocadas entre o gerente, os agentes e a UPS, para permitir análise posterior dos eventos e das mensagens de erro relatadas pelos agentes. Este arquivo deve ser armazenado em meio não volátil e confiável, para que as informações possam ser recuperadas mesmo com falhas permanentes.

O módulo de tomada de decisão utiliza as informações sobre a UPS, a política de gerência e o estado atual dos equipamentos para atuar nos agentes.

O módulo de comunicação com o reserva é encarregado de enviar mensagens regulares indicando o bom funcionamento do gerente. Também é utilizado para replicar informações de *log* no reserva.

5.2 Gerente Reserva

O gerente reserva recebe cópias das mensagens enviadas pelo primário. Em caso de falhas do primário, o reserva assume seu lugar na gerência da rede. Possui a mesma estrutura do gerente principal e também armazena a política de gerência.

Durante o funcionamento normal do sistema, apenas recebe mensagens do primário e as armazena no arquivo de *log*. Se não for contactado dentro de certo período de tempo, tenta contactar o primário. Caso não consiga, assume a gerência da rede.

O reserva não precisa receber as notificações de troca de estado dos agentes pois, caso assuma a gerência, deverá contactar os agentes para realizar a troca do endereço do gerente, e estes informarão seus estados.

5.3 Agentes

Os agentes recebem informações do sistema operacional sobre as transições no estado de consumo dos equipamentos, que ocorrem de acordo com a política de gerência utilizada localmente, e as repassam ao gerente. Em situações de operação emergencial com baterias, o agente recebe comandos do gerente e repassa ao sistema operacional para que o sistema seja colocado em estado de baixo consumo.

O agente possui arquitetura simples, pois não pode sobrecarregar o nodo gerenciado. Simplesmente realiza a tradução das mensagens do gerente para o sistema operacional e do sistema operacional para o gerente. O agente também deve fornecer avisos para o usuário local sobre as medidas de redução de consumo que serão executadas no equipamento.

O agente possui dois módulos: um agente SNMP e outro que consiste de uma biblioteca de funções para atuação no sistema, de acordo com a figura 5.3.

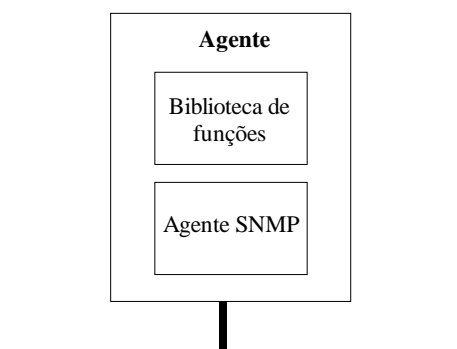


FIGURA 5.3 - Composição do agente

A biblioteca deve implementar dois tipos de funções:

- funções de atuação na MIB;
- função de notificação de mudança no estado de consumo dos dispositivos.

O gerente determina o comportamento do agente, alterando os valores dos objetos da MIB. Para isto, deve enviar mensagens que serão transformadas em chamadas às funções da biblioteca.

A biblioteca deve ser executada quando o agente iniciar, para que seja registrada junto ao sistema operacional, de maneira a ser informada sobre transições no estado de consumo dos dispositivos.

A implementação do agente é dependente do ambiente em que será executado. Deve ser considerado o *hardware*, o padrão de gerência e o sistema operacional utilizados e, para cada combinação destes, deve ser desenvolvido um agente específico.

Em função da análise dos padrões de gerência, recomenda-se que sejam utilizados computadores compatíveis com ACPI, que permite maior controle sobre o *hardware*, e sistemas operacionais Windows 98, Windows 2000 ou (futuramente) Linux, pois são os sistemas possuem suporte ao ACPI.

5.4 Sinal de Vida

O gerente cria a visão do sistema em função das mensagens de troca de estado enviadas pelos agentes. No entanto, se o equipamento for abruptamente desligado, nenhuma mensagem será enviada ao gerente. Por esse motivo, os agentes dos equipamentos em estado ativo enviam mensagens periodicamente ao gerente, mesmo que não haja mudança no estado. Os equipamentos que estejam em estado de baixo consumo não enviam esta notificação. Como os agentes estarão programados para enviar este sinal com o mesmo intervalo de tempo, por exemplo t minutos, então a cada t minutos, o gerente fica atualizado sobre os equipamentos que estão em estado ativo.

Equipamentos que constem como ativos na tabela de visão mas não enviem sinal de vida dentro do tempo t , são considerados em estado desconhecido.

5.5 Estados de Consumo das Entidades

Os estados de consumo dos equipamentos variam em função do padrão de gerência utilizado pelo fabricante. Por esse motivo, o sistema de gerência possui seus próprios estados (que são definidos pelo administrador da rede) e estes são mapeados nos estados reais suportados pelos dispositivos. Os estados suportados pelo gerente são descritos na MIB definida (seção 5.11).

O protótipo implementado comporta quatro estados. Toda entidade possui no mínimo dois estados possíveis: **ligado** (*on*) ou **desligado** (*off*). Em situações de falhas, pode não ser possível detectar o estado momentâneo da entidade, o que origina o estado **desconhecido** (*ignored*). Este é um estado que possui relevância apenas para o gerente, sem contrapartida na entidade. Quando este estado é atribuído a um equipamento, na verdade ele se encontra em algum dos outros, mas não foi possível determiná-lo momentaneamente (seção 5.9).

É interessante que as entidades possam ser colocadas em estados de baixo consumo. Assim, o sistema comporta um estado de baixo consumo ou de **espera** (*sleep*). Embora um equipamento ou dispositivo possa ter mais de um estado de espera, cabe ao agente escolher o estado apropriado a que a entidade deve ser levada.

Outros estados de consumo diferenciados podem ser definidos, caso seja adequado. Para isso, apenas a tabela de estados, no arquivo de configuração, deve ser atualizada de forma a conter o novo estado e sua interpretação para cada equipamento gerenciado (seção 6.3.1).

5.6 Comunicação entre Gerente e Agentes

Normalmente utiliza-se *polling* para gerenciar redes com SNMP [BLA92]. O *polling* consiste em deixar a iniciativa da obtenção de informações para o gerente, que deve periodicamente consultar os agentes da rede. No entanto, esta estratégia não é

apropriada para gerência de energia pois, para atender ao *polling*, o sistema deve estar funcional, possivelmente causando alteração no estado de energização.

Outra estratégia, chamada de interrupção, ou notificação de eventos, é então utilizada. Nessa estratégia, os agentes informam ao gerente a ocorrência de eventos de interesse. Dessa maneira, é possível aos agentes informarem mudanças no estado de energização antes que o sistema entre em estado de baixo consumo. Esta estratégia também aumenta a escalabilidade do sistema, pois evita que o gerente realize requisições desnecessárias. Outro fator positivo é que o gerente sempre possui informações atualizadas sobre o estado dos equipamentos gerenciados.

5.7 Interface Gráfica

A interface gráfica possui duas partes. Uma parte permite a visualização do estado de consumo momentâneo de cada equipamento gerenciado e permite enviar comandos para estes equipamentos. Também permite visualizar o arquivo de *log* do gerente. A segunda parte permite ao administrador determinar a política de gerência (quais equipamentos serão gerenciados e a prioridade de cada um) a ser implantada pelo gerente.

A estrutura modular do gerente permite que este funcione sem interface gráfica, visto que, uma vez definidas as políticas de gerência, a interface possui papel principalmente informativo para o administrador.

5.8 Modelo de Falhas

O modelo de falhas considerado é identificado a partir dos componentes modulares da rede e compreende falhas no gerente, nos agentes ou na rede que os interliga²⁰. Assume-se que os nodos possam apresentar falhas de colapso (*crash*) [JAL94], sendo permitida a recuperação dos processos gerente ou agente. Assume-se que, se o equipamento está livre de falhas, o processo correspondente (gerente ou agente) também está funcionando. Os equipamentos não podem ser remotamente gerenciados sem o UDP/IP estar carregado. Esta restrição possui implicações na confiabilidade do sistema, conforme explicado adiante. Não é exigido o sincronismo entre os relógios dos equipamentos.

O gerente pode apresentar falhas de curta ou de longa duração. Falhas de curta duração, ou transitórias, são tratadas de maneira transparente pelo próprio gerente. A título de exemplo, falhas temporárias podem ser devidas à reinicialização da máquina em que o gerente é executado. Falhas de longa duração do gerente, ou permanentes, são detectadas pelo reserva, que assume a gerência. O dispositivo de armazenamento não volátil do gerente (disco rígido) deve ser estável, ou seja, deve permitir a recuperação de

²⁰ No sistema considerado, a subdivisão de menor interesse são os componentes da rede. Assim, assume-se que, as falhas ocorrem nestes componentes, mas a solução tenta evitar defeitos no fornecimento e controle de energia na rede - deste ponto de vista ela é tolerante a falhas.

dados mesmo após a ocorrência de falhas permanentes, para posterior análise pelo administrador.

Os agentes podem falhar e recuperar-se assincronamente, e sua recuperação não exige nenhuma ação especial dos componentes do sistema, já que na inicialização os agentes enviam informação sobre seu estado para o gerente.

A rede pode apresentar falhas físicas (componentes da rede) ou lógicas (expiração de *timeouts*), e se recuperar. Em caso de ocorrerem partições, a partição que contiver o gerente será gerenciada. As mensagens transmitidas pela rede podem ser perdidas, duplicadas ou reordenadas, mas não alteradas (comportamento bizantino).

5.8.1 Falhas do Gerente

Para lidar com falhas longas do gerente, foi escolhida a técnica de réplica ou cópia primária (*primary site*) [JAL94, MUL93], também conhecida por primário-reserva (*primary-backup*), como modelo para a implementação do mesmo. Nesta técnica, os dados são replicados em $k+1$ nodos e o sistema suporta falha em até k nodos, exigindo que falhas de nodo sejam diferenciadas de falhas que resultem em partições da rede. Um destes nodos é o primário e os outros são os *backups* ou reservas. Neste sistema, os dados que devem ser replicados são as informações sobre o estado de energização dos equipamentos da rede e as políticas de gerência. Todas as operações com os clientes são realizadas pelo primário, que se encarrega de repassar as alterações de estado aos reservas. Quando k reservas falharem, o sistema deixará de funcionar. Se até k reservas falharem, nenhuma ação precisará ser realizada. Se o primário falhar, um novo primário será escolhido entre os reservas. Na prática entretanto, um reserva é suficiente.

A estratégia adaptada ao modelo permite falha permanente no gerente, utilizando um gerente primário e outro reserva. Ambos possuem as políticas de gerência. Todas as operações são realizadas pelo gerente primário e os agentes conhecem apenas o endereço deste. Ao receber notificações da UPS e sempre que enviar comandos, estas mensagens referentes aos comandos são replicadas no reserva, que as armazena em seu arquivo de *log*. Ao detectar falha do gerente, o reserva envia alarme para o administrador e assume a gerência, analisando seu arquivo de *log* para começar a gerência a partir do ponto em que a comunicação cessou.

O reserva tenta contatar o gerente caso não receba mensagens dentro de certo limite de tempo. Se não houver resposta do gerente, o reserva assume a gerência do sistema, gerando alarme para o administrador, enviando aos agentes seu próprio endereço como sendo gerente e contatando a UPS para saber a situação de fornecimento de energia. Para isso, o reserva possui uma senha especial que permite alterar o endereço do gerente na MIB dos agentes²¹. Além disso, o novo gerente deve tentar contatar o antigo constantemente, para detectar sua recuperação.

Esta técnica foi escolhida pelos seguintes motivos:

²¹ Essa segurança apenas é possível se for utilizado SNMPv3. Outras versões do protocolo utilizam apenas os identificadores de comunidade.as *community names*

- garante serialidade²², pois apenas o gerente primário trata as mensagens;
- permite que os agentes possuam apenas um endereço para contato com o gerente, simplificando sua implementação;
- economiza banda da rede, pois o gerente pode condensar várias mensagens e mandá-las de uma só vez para o *backup*;
- possibilita que sistemas menos críticos utilizem apenas uma máquina configurada como gerente;
- utiliza poucas mensagens e portanto causa pequena sobrecarga ao sistema.

Para lidar com falhas curtas dos agentes, o gerente utiliza o sinal enviado pelos agentes em sua inicialização sem necessidade de troca de mensagens extras. Ao iniciar, o gerente verifica se está iniciando uma sessão de gerência ou se está se recuperando de uma falha através de consulta ao arquivo de *log*. Se estiver se recuperando de falha, dentro de, no máximo, *t* minutos, o gerente saberá quais equipamentos estão em atividade na rede, devido ao sinal de vida. Os outros equipamentos são considerados em estado desconhecido e são gerenciados como se estivessem em estado de baixo consumo.

5.8.2 Falhas dos Agentes

A recuperação dos agentes não exige nenhuma atividade extra, já que, ao ser inicializado, o agente informa ao gerente o estado de energização atual (ligado) e este atualiza a visão do sistema.

5.9 Segurança

A segurança é um ponto muito importante neste tipo de sistema. A utilização indevida dos mecanismos de controle de consumo pode ocasionar falhas de funcionamento dos equipamentos (ataques de DoS²³) ou danos ao *hardware*. O ponto mais apropriado para inserir mecanismos de segurança é no protocolo SNMP. A versão 3 deste protocolo disponibiliza um serviço seguro, mesmo se as mensagens tiverem que trafegar por redes não confiáveis; algumas características são apresentadas nessa seção. No entanto, por estar em fase de padronização, ainda não existem ferramentas gratuitas disponíveis, portanto o protótipo implementado neste trabalho utiliza o SNMPv2c.

Usuários comuns não podem alterar o estado de consumo de equipamentos remotos, pois o sistema tornar-se-ia alvo fácil a ataques de negação de serviço. Por outro lado, o sistema deve garantir que os equipamentos se submetam às determinações do administrador, evitando a utilização indevida dos recursos de energia disponíveis. A segurança do modelo é implementada nos pontos a seguir especificados.

²² Propriedade desejável em sistemas distribuídos que garante que o resultado da execução concorrente de ações é o mesmo obtido pela execução sequencial dessas ações [MUL93].

²³ DoS – *Denial of Service* ou ataque de negação de serviço.

As mensagens do gerente devem ser autenticadas pois os agentes apenas executam comandos do gerente que estiver registrado em sua base de informações. Comandos recebidos de gerentes não registrados são considerados tentativas de operação indevida.

As mensagens dos agentes devem ser autenticadas para evitar notificações falsas de desligamento de equipamentos.

A troca do endereço do gerente pode apenas ser realizada pelo administrador ou pelo gerente reserva. Para isso é escolhida uma senha especial que dá direito a executar esta operação no agente. A senha deve ser encriptada quando for utilizada.

As mensagens dos agentes e do gerente devem ser encriptadas para evitar que um intruso monte uma visão do sistema sabendo o estado de energização dos equipamentos.

O usuário não pode ter acesso a mecanismos que inibam a gerência de energia. Por exemplo, computadores devem possuir senha de acesso ao BIOS para evitar que a gerência de energia seja desabilitada.

A ocorrência de falhas nos agentes deve ser investigada pelo gerente para determinar se estas não foram induzidas por um usuário. O gerente deve determinar se a pilha UDP/IP do equipamento está carregada²⁴. Se estiver, considera-se que o usuário deliberadamente provocou a falha²⁵ interrompendo a execução do *software* agente.

5.10 Integração com Plataformas de Gerência

É bastante comum o uso de plataformas de gerência em ambientes de rede. Estas plataformas consistem de diversos aplicativos de gerência que compartilham o mesmo ambiente de execução e informações sobre a rede gerenciada, fornecendo uma interface comum para interação com o administrador. Estas plataformas podem ser estendidas pois também fornecem um substrato comum e diversas APIs de funções, que podem ser utilizadas para o desenvolvimento de novas aplicações de gerência. Alguns exemplos de plataformas encontradas no mercado são o HP OpenView, o Sun NetManager e o IBM Tivoli.

Embora sejam bastante úteis para a gerência, estas plataformas possuem duas desvantagens para utilização em determinados ambientes. A primeira é seu custo, que pode tornar proibitiva a aquisição de uma ferramenta destas para alguns ambientes. A segunda desvantagem é a complexidade, que exige equipamentos com recursos computacionais poderosos que também acabam possuindo custo elevado. Muitas vezes, para gerenciar certos aspectos específicos das redes, não é necessário utilizar uma plataforma complexa.

²⁴ Através do programa *ping*, por exemplo.

²⁵ Conforme exposto no modelo de falhas, por hipótese, os agentes apenas podem falhar se o equipamento em que residem apresentar falha.

A decisão sobre a aquisição ou a não aquisição de uma destas plataformas define a necessidade de desenvolver diferentes versões do *software* de gerência, uma para ser integrada com uma plataforma de gerência ou outra para uso isolado. A implementação apresentada neste trabalho foi desenvolvida para uso isolado. No entanto, foi estudada a integração com uma plataforma de gerência, o HP OpenView [HEW97, HEW97a]. A integração com outras plataformas deve seguir os mesmos princípios.

Esta plataforma disponibiliza serviços de cinco tipos: são serviços de comunicação, de gerência de dados, de interface com o usuário, de manipulação de eventos e de ajuda.

Os serviços de comunicação fornecidos são: envio e recebimento de informações e o recebimento de notificações dos agentes. Eles utilizam o protocolo SNMP.

Os serviços de gerência de dados permitem que os diversos aplicativos de gerência possam compartilhar dados dos equipamentos gerenciados tais como nome, endereço, topologia e outros.

Os serviços de interface com o usuário permitem que as aplicações de gerência que executam na plataforma possuam uma interface uniforme para comunicação com o administrador. O HP Openview disponibiliza APIs que possuem componentes gráficos como menus, barras de tarefas, mapas, símbolos e gráficos.

Os serviços de manuseio de eventos permitem que as aplicações sejam notificadas da ocorrência de eventos de teclado, do sistema operacional, dos agentes SNMP ou de outros aplicativos de gerência. Para utilizar os serviços de eventos, a aplicação deve implementar rotinas de *callback* e registrar essas funções junto ao laço de eventos do *software*.

Os serviços de ajuda permitem que as aplicações ofereçam ao usuário um sistema de documentação unificado e organizado, os quais utilizam uma estrutura de hipertexto.

Estes serviços são disponibilizados através da API de funções da plataforma. As aplicações devem ser compiladas e ligadas (*linked*) às bibliotecas do OpenView. Após, a aplicação deve ser registrada junto aos arquivos de configuração do OpenView, descrevendo como integrar a aplicação na estrutura de menu, como a aplicação será executada, como gerenciar processos da aplicação e onde as informações de ajuda estão localizadas.

Para que o gerente possa interagir com plataformas de gerência, deve ser desenvolvido um módulo específico para cada plataforma, que realize a interface entre os pontos de comunicação do gerente e as bibliotecas de funções da plataforma.

No caso do OpenView, o módulo deve permitir ao gerente utilizar os serviços de comunicação da plataforma, enviando ao OpenView as mensagens do SNMP que devem ser enviadas aos agentes e recebendo destes as interrupções enviadas ao gerente. Este módulo também será responsável por utilizar os serviços de gerência de dados para obter informações como endereço dos equipamentos, junto ao OpenView. Este módulo

deve também receber informações sobre eventos, registrando as rotinas de *callback* necessárias junto ao OpenView.

O gerente pode utilizar os serviços de interface e de ajuda do OpenView, desde que um módulo de interface apropriado seja desenvolvido.

5.11 Base de Informações de Gerência

Para trocar informações, tanto o gerente como os agentes implementam uma base de informações de gerência comum.

A MIB NetPower é derivada da MIB para controle de consumo em ambiente de redes de computadores [KRO99a], mostrada na figura 5.4. Essa MIB foi proposta no período inicial do estudo sobre controle de consumo e define três grupos: um para equipamentos com implementação do APM, um para equipamentos com implementação do ACPI e um para equipamentos com mecanismos próprios de controle de consumo. Cada grupo possui um objeto para controlar o estado do sistema (*systemStatus*), do processador (*procStatus*) e dos dispositivos (*devGenTable*, *devAPMTable* e *devACPITable*). O grupo genérico contém ainda uma tabela para a descrição dos estados de consumo implementados (*stTable*).

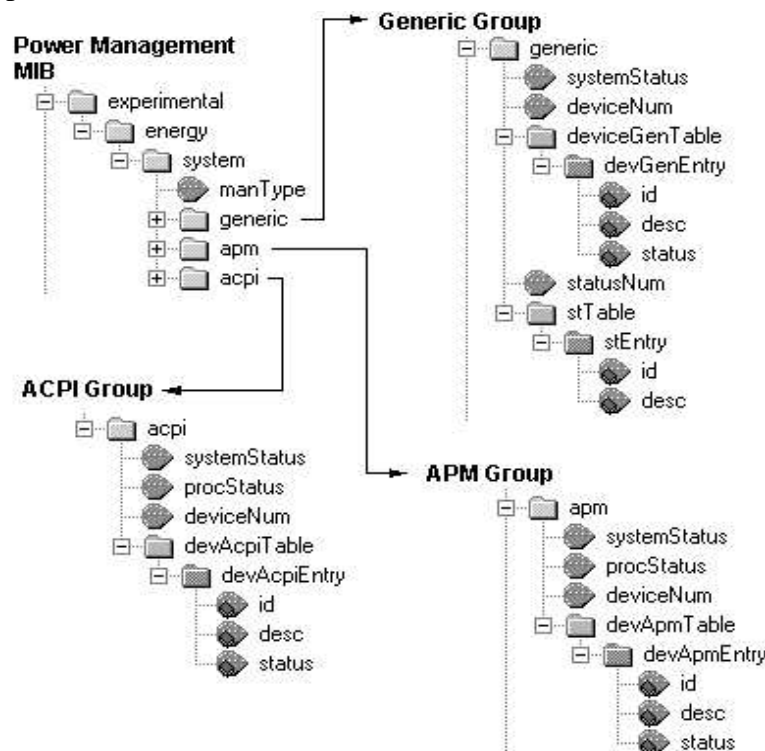


FIGURA 5.4 - MIB para controle de consumo

Para o sistema NetPower, foi necessário que uma MIB específica fosse desenvolvida, eliminando alguns objetos redundantes da MIB anterior e acrescentando novos objetos específicos para o gerente NetPower. Assim, a MIB NetPower foi proposta: seus objetos são mostrados na tabela 5.1 e sua estrutura é mostrada na figura 5.5.

TABELA 5.1 - Objetos da MIB proposta

<i>Objeto</i>	<i>Descrição</i>
<i>mibVersion</i>	versão da MIB implementada
<i>pwrStd</i>	padrão de gerência utilizado no equipamento
<i>pwrStdVersion</i>	versão do padrão de gerência
<i>mgrAddr</i>	endereço do gerente para quem as interrupções (<i>traps</i>) devem ser enviadas e de quem os comandos devem ser aceitos
<i>sndTrp</i>	valor lógico que indica se o agente deve ou não enviar interrupções informando mudança de estado. Esta característica pode ser desativada pelo gerente, quando este for encerrado
<i>msg</i>	mensagem a ser mostrada ao usuário
<i>cmd</i>	cadeia de caracteres que indica o caminho e o nome de um <i>script</i> ou programa a ser executado pelo agente
<i>devTable</i>	tabela que contém os dispositivos gerenciados. Possui, no mínimo, uma entrada que representa o sistema
<i>devEntry</i>	entrada da tabela
<i>devId</i>	identificação do dispositivo
<i>devDesc</i>	descrição do dispositivo
<i>devState</i>	estado de consumo do dispositivo
<i>devConsumption</i>	consumo instantâneo de energia do dispositivo

Todos os objetos armazenam valores correspondentes a conjuntos de caracteres (*strings*), à exceção dos objetos *sndTrp*, do tipo lógico, e *devConsumption*, que é do tipo numérico.

Os objetos implementados na MIB visam identificar o ambiente em que o agente é executado (*mibVersion*, *pwrStd*, *pwrStdVersion*, *mgrAddr*), receber comandos do gerente (*sndTrp*, *msg*, *cmd*), descrever, atuar nos dispositivos gerenciados (*devTable*) e informar alterações no estado de energização de dispositivos (*chgTable*).

A utilização de cadeias de caracteres para os campos *devId* e *devState* permite que, no futuro, outros dispositivos possam ser gerenciados e que possam ser utilizados outros valores para os estados de consumo. Para isso, é necessário que o gerente envie a identificação do dispositivo e o novo estado, e que o agente esteja preparado para lidar com esses valores, não sendo necessário alterar a MIB.

Essa MIB pode ser estendida no futuro e o objeto *mibVersion* permite que o gerente saiba com antecedência que objetos a MIB contém e quais operações são permitidas.

O objeto *devState* indica o estado em que o dispositivo se encontra, em operações *get*, ou o estado em que deve ser colocado, em operações *set*. Em operações

set, o valor colocado nesse objeto indica também a maneira como a transição será realizada. A tabela 5.2 mostra os valores definidos para esse objeto.

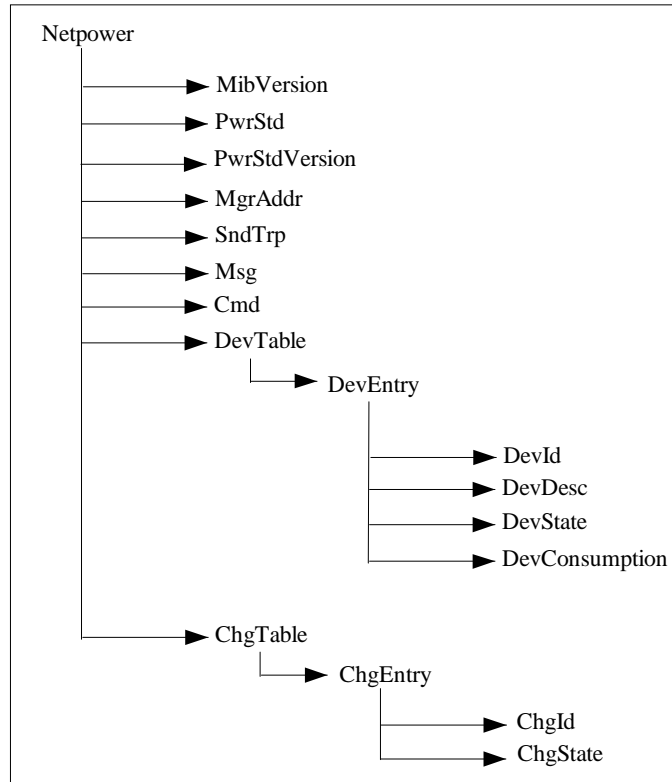


FIGURA 5.5 - MIB NetPower

Alguns valores possuem significado apenas para instruções *get* e outros apenas para instruções *set*. Por exemplo, para uma instrução *get*, o retorno de qualquer valor indica que o dispositivo está em estado de espera (*sleep_moretime*, *sleep_context* ou *sleep_force*). O valor *inactive* faz sentido apenas para instruções *get*.

Ao receber os valores *sleep_moretime* ou *off_moretime*, o agente pode automaticamente tomar a decisão de colocar ou não o dispositivo nesses estados, avaliando a atividade do dispositivo no momento ou consultando o usuário, se o equipamento for de utilização interativa.

Os agentes não precisam implementar todos os valores para todos os dispositivos. Por exemplo, para desligar o monitor, o estado *off_context* não faz sentido.

TABELA 5.2 - Estados dos dispositivos

Valor	Significado
<i>active</i>	o dispositivo está completamente energizado e ativo
<i>active_idle</i>	o dispositivo está plenamente energizado mas ocioso
<i>sleep_moretime</i>	o dispositivo deve ser colocado em estado de espera se não estiver realizando trabalho

<i>Valor</i>	<i>Significado</i>
<i>sleep_context</i>	o dispositivo deve ser colocado em estado de espera e seu contexto deve ser salvo
<i>sleep_force</i>	o dispositivo deve ser colocado em estado de espera mesmo que o contexto não possa ser salvo
<i>off_moretime</i>	o dispositivo deve ter sua energia cortada se não estiver realizando trabalho
<i>off_context</i>	o dispositivo deve ter sua energia cortada e seu contexto deve ser salvo
<i>off_force</i>	o dispositivo deve ter sua energia cortada mesmo que o contexto não possa ser salvo
<i>reset</i>	reinicializa o dispositivo
<i>logout</i>	encerra a seção do usuário
<i>ignored</i>	não é possível determinar o estado do dispositivo

Além desses objetos, foram definidas interrupções para que os agentes informem transições na situação de consumo ao gerente. As interrupções definidas são:

- entrada no estado ativo – o agente envia essa interrupção quando o equipamento é ligado ou quando retorna de um estado de espera;
- entrada em estado de espera – é enviada quando o equipamento entra em estado de baixo consumo;
- desligamento – é enviada quando o equipamento está prestes a ser desligado;
- alteração de estado em dispositivo – é enviada sempre que o estado de consumo de algum dispositivo é alterado. Além de enviar essa interrupção, o agente deve inserir uma entrada na tabela *chgState*, colocando a identificação do dispositivo e o estado;
- sinal de vida – é enviada a regularmente ao gerente para indicar que o agente continua em estado ativo.

6 Descrição do Gerente

O gerente é o componente principal do modelo de gerência e será apresentado no decorrer deste capítulo. O projeto do gerente objetiva:

- a utilização de qualquer modelo de UPS;
- controle de diversos tipos de dispositivos, podendo ser adaptado à evolução da tecnologia para controlar novos dispositivos;
- portabilidade, de modo que versões do gerente e dos agentes possam ser facilmente implementadas em diversas plataformas, e que estas versões possam interoperar através da interface de comunicação entre os módulos do gerente.

Sua estrutura básica é apresentada na figura 6.1

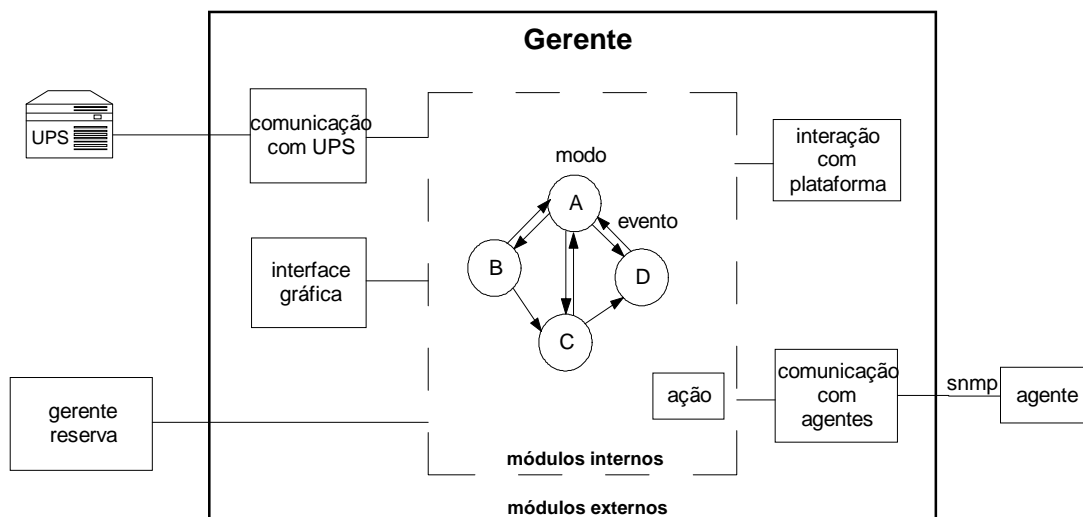


FIGURA 6.1 - Estrutura do gerente

O gerente é composto de um bloco de *software* que contém os módulos internos e alguns módulos externos que se comunicam com os módulos internos através de *sockets*.

Internamente ao gerente, os módulos trabalham com as seguintes entidades: **eventos**, **modos de operação** e **entidades gerenciadas**. A figura 6.2 mostra o relacionamento destes elementos. As setas indicam a ocorrência de eventos.

A ocorrência de **eventos** de energia significa que houve mudança no fornecimento. Ao receber a informação da ocorrência de um evento, o gerente altera o modo de operação do sistema.

Os **modos de operação** determinam quais entidades gerenciadas podem consumir energia ou quais não devem continuar consumindo. Esses modos são mais ou menos restritivos em função das definições do administrador. Ao entrar em um modo de

operação, o gerente dispara uma série de ações às entidades gerenciadas para adequar o consumo da rede.

As **entidades gerenciadas** são equipamentos e seus dispositivos internos.

Os modos de operação e os eventos formam uma máquina de estados em que qualquer estado pode ser inicial ou de saída, conforme a figura 6.2.

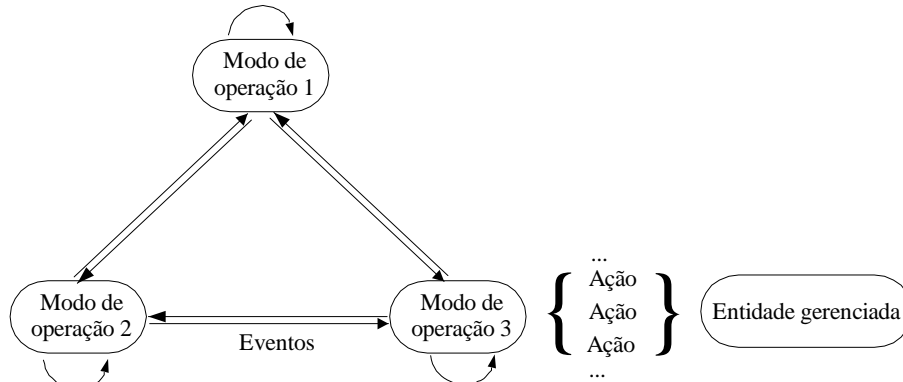


FIGURA 6.2 - Relações entre modos de operação, eventos e entidades gerenciadas

6.1 Composição

O gerente é dividido em módulos, que podem ser internos ou externos. Os módulos internos (delimitados pelo retângulo interno tracejado na figura 6.1) são implementados como um bloco único de *software*. Os módulos externos são implementados como blocos de *software* independentes que se comunicam com o bloco principal através de *sockets*.

A interface de *sockets* foi escolhida por ser um meio de comunicação entre processos implementado em diversas plataformas e que permite inclusive interoperabilidade, ou seja, um módulo pode ser executado em uma plataforma e outro módulo em outra. Além disso, a interface de *sockets* permite boa separação entre as estruturas dos módulos, estabelecendo uma interface bem definida para a comunicação, é fácil de ser programada e é bastante flexível. Todavia, aspectos de segurança devem ser considerados quando a interface de *sockets* é utilizada por esta ser implementada sobre o protocolo TCP/IP. As soluções para evitar problemas de segurança são a restrição à recepção apenas de *sockets* originados a partir da máquina local (*Unix Domain Sockets*)²⁶, ou a implementação de túneis seguros entre os computadores que executam os diversos módulos. Esses túneis podem ser facilmente implementados com as tecnologias SSL (*Secure Socket Layer*) ou SSH (*Secure Shell*).

A estratégia de separação em blocos objetiva a construção de um gerente que possa ser utilizado em qualquer ambiente de computação sem necessidade de reescrita do bloco principal. Apenas os módulos externos devem ser implementados para adaptação ao ambiente desejado. Os módulos externos do gerente são os seguintes:

²⁶ Existem bibliotecas que emulam esse mecanismo de comunicação entre processos no Windows utilizando outros mecanismos existentes nesse sistema operacional.

- monitor de UPS;
- interface (gráfica ou simples);
- monitor de agentes;
- interação com plataforma de gerência.

O módulo **monitor de UPS** é responsável por informar ao gerente a situação do fornecimento de energia. Deve ser construído um módulo desses para cada modelo de UPS que possa fazer parte do sistema e um gerente pode ter diversos módulos desses simultaneamente. Esse módulo pode ser executado no mesmo computador que o gerente ou em um computador remoto, podendo também ser desenvolvido pelo fabricante da UPS. O protocolo utilizado é bastante simples, consistindo de uma cadeia de caracteres (*string*) enviada do módulo para o gerente, informando a ocorrência de um evento. A cadeia de caracteres enviada por esse módulo deve estar presente no **arquivo de configuração** do gerente, na seção **events** (seção 6.3).

A **interface** possui duas funções. A primeira é a definição de parâmetros de funcionamento do gerente como: configuração do gerente (porta de *socket* a ser monitorada à espera de eventos, quantidade de informações de *log*, entre outros), descrição do ambiente a ser gerenciado (equipamentos gerenciados, definição de grupos, entre outros), e determinação do comportamento do gerente na ocorrência de eventos de energia (ações a serem executadas nos agentes). A segunda é a interação com o administrador, mostrando o estado de consumo de cada equipamento da rede e a situação da UPS e enviando os comandos do administrador ao gerente. O uso da interface para interação com o administrador é opcional. Uma vez definidos os parâmetros de funcionamento, o gerente pode ser executado de forma não interativa. Quando necessário, o administrador pode conectar uma interface e desconectá-la a qualquer momento.

A implementação como um módulo permite que possam ser desenvolvidas interfaces simples (estilo linha de comando, por exemplo) ou elaboradas (estilo interfaces gráficas, por exemplo). Uma possibilidade interessante é o desenvolvimento de uma interface acessível através da *Web*, permitindo ao administrador atuar no sistema de qualquer lugar, conectado via Internet.

O módulo **monitor de agentes** permite a interação com os agentes da rede. Sua implementação como módulo objetiva afastar o gerente de detalhes de versão do protocolo a ser utilizado e permitir que possa ser utilizado o serviço de comunicação de uma plataforma de gerência, se for desejado.

O módulo de **interação com plataforma de gerência** é responsável pelo acesso aos serviços da plataforma de gerência, descritos na seção 5.10, quando esta for utilizada. Na presença de plataforma de gerência, o gerente requisita a este módulo informações sobre o ambiente gerenciado (serviço de gerência de dados); este módulo utiliza as interfaces de comunicação por *sockets* do gerente para tratar eventos de interface e de agentes (serviços de comunicação e de interface). Este módulo também deve tratar informações sobre a ocorrência de eventos e implementar as funções de *callback* da plataforma de gerência (serviço de manipulação de eventos).

A comunicação com o gerente reserva permite que o sistema seja tolerante a falhas do gerente, já que este se tornaria um ponto único de falhas (*single point of failures*).

Para utilizar eficientemente os recursos computacionais, o bloco principal é implementado como um processo *multithreads*. Durante a operação do gerente sete *threads* são executadas nesse módulo:

- *thread* principal, que inicializa as estruturas de dados do gerente, dispara as outras *threads* e, a partir desse momento, manipula as respostas ao eventos monitorados;
- *thread* de monitoração de eventos de UPS;
- *thread* de monitoração de eventos de interface;
- *thread* de monitoração de eventos de agentes;
- *thread* de monitoração de eventos de gerente reserva;
- *thread* de monitoração de eventos de plataforma de gerência;
- *thread* de geração de *log*.

O bloco principal, de acordo com a figura 6.3, é composto dos seguintes módulos: **inicialização**, **polling**, **interação** (com UPS, com interface, com agentes, com gerente reserva e com plataforma de gerência), **máquina de estados**, **interpretador de estados**, **interpretador de ações**, **relógio** e **visão**.

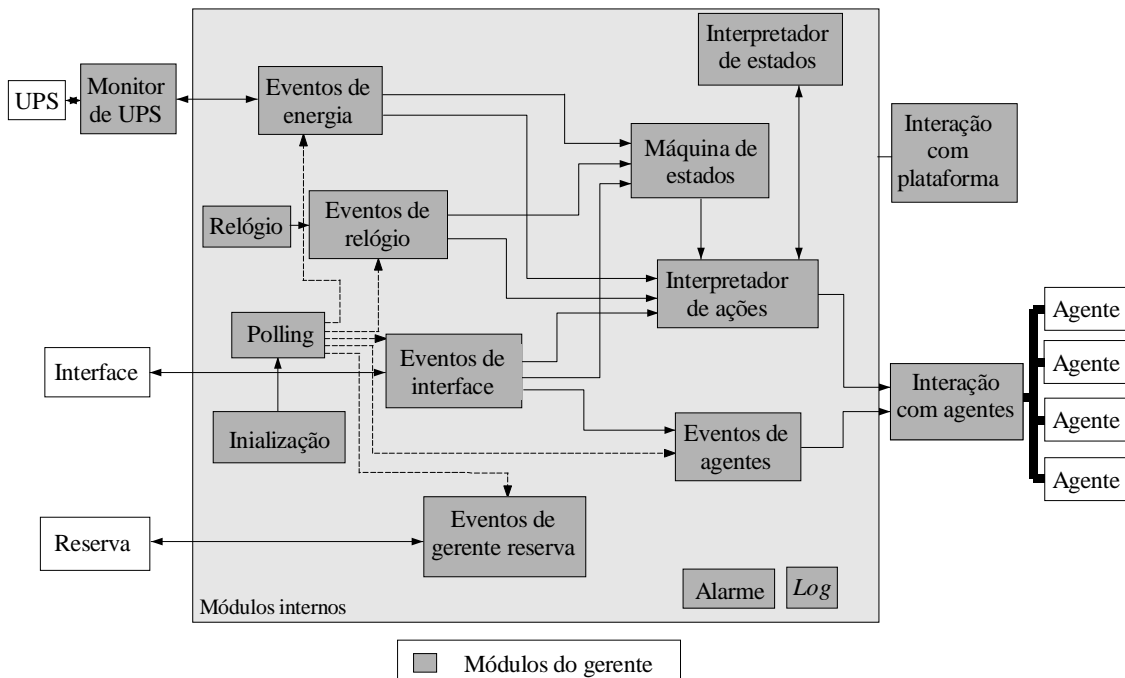


FIGURA 6.3 - Estrutura do gerente

O **módulo de inicialização** é responsável por ler os arquivos de inicialização do gerente, que contêm informações de configuração e de comportamento. Estes arquivos devem ter sido gerados anteriormente pelo administrador, através da interface gráfica. Este módulo também deve disparar as outras *threads* do gerente.

O **módulo de *polling*** deve pesquisar os diversos módulos de interação para verificar a ocorrência de eventos e disparar as ações necessárias.

Os **módulos de interação** devem monitorar as interfaces de *sockets* e registrar a ocorrência de eventos para que o módulo de *polling* execute as ações necessárias. Cada módulo de interação possui uma *thread* de execução.

A **máquina de estados** é responsável por programar as ações que serão disparadas, de acordo com a ocorrência de eventos. Este módulo possui uma lista dos possíveis eventos que a UPS pode relatar e uma lista de ações a disparar para cada evento.

O **interpretador de estados** contém uma tabela responsável por traduzir os estados do sistema de gerência nos estados suportados pelo equipamento gerenciado. Para essa tradução, leva em consideração a arquitetura do equipamento, o padrão de gerência de energia utilizado, o sistema operacional e o dispositivo.

O **interpretador de ações** realiza a tradução entre as ações abstratas definidas pelo administrador (ligar ou desligar dispositivo) em ações do tipo *get* e *set* para serem enviadas aos agentes.

O **módulo de relógio** deve gerar pulsos a intervalos regulares para que o interpretador de ações execute as ações no momento desejado.

O **módulo de visão do sistema** é responsável por manter uma lista dos equipamentos gerenciados e seus dispositivos internos, com o estado de consumo atual de cada um. Sempre que uma ação for disparada ou um evento de agente for detectado, este módulo atualiza a visão do sistema. Essa visão é útil para o acompanhamento da execução das ações pelos agentes e para informar à interface gráfica o estado dos equipamentos da rede.

O gerente é um *software* orientado a eventos, que podem ser: **de energia, de relógio, de interface, de agentes, de gerente reserva** ou de **plataforma de gerência**. Desses, apenas os eventos de energia, de relógio e de interface ocasionam ações enviadas aos agentes. Os eventos de agentes, de gerente reserva e de plataforma de gerência são tratados internamente ao gerente.

Os **eventos de energia** são relacionados a informações obtidas junto à UPS. A ocorrência de um evento desse tipo deve provocar mudança no estado de consumo da rede de modo a adequar o consumo às determinações do administrador.

Os **eventos de relógio** servem para que o interpretador de ações dispare as ações contidas na lista de ações no momento programado.

Os **eventos de interface** podem atuar nos módulos eventos de energia, interpretador de ações e visão do sistema. Também podem obter dados das estruturas de dados internas (lista de ações, por exemplo).

Os **eventos de agentes** são interrupções (*traps* do SNMP) enviadas pelos agentes que indicam mudança de estado de consumo em algum dispositivo dos

equipamentos gerenciados. Ao receber esses eventos o gerente deve pesquisar o agente em questão para descobrir qual dispositivo teve o estado alterado.

Os **eventos de gerente reserva** indicam que o gerente deve enviar um sinal de vida ao reserva. Caso isso não aconteça, o reserva irá assumir a gerência mudando o valor do objeto *mgrAddr* na MIB dos agentes (seção 5.11) para o seu próprio endereço.

Os **eventos de plataforma de gerência** podem atuar em diversos módulos internos. Esse eventos podem ser de interface, de agentes ou de energia.

6.2 Funcionamento

A definição das políticas de gerência deve ser realizada previamente pelo administrador, visto que as redes são compostas de diferentes modelos de equipamentos e possuem diferentes necessidades de serviços. Assim, o administrador é o elemento que possui melhor conhecimento dos recursos disponíveis e das necessidades da rede e está em melhor situação para definir as políticas de gerência.

Inicialmente o administrador deve definir a configuração e o comportamento do gerente através da interface, que acaba gerando dois arquivos: um de configuração e outro de comportamento²⁷. A utilização de arquivos garante que as informações de gerência sejam armazenadas em meio não volátil. O gerente reserva deve receber cópia desses arquivos.

A diferenciação na execução entre gerente e reserva ocorre logo no início. O reserva entra em um laço no qual são feitas tentativas de contato como gerente em intervalos de tempo. Essa atividade apenas se modifica quando ocorre falha que interrompe a comunicação. Nesse caso, o reserva executa o **módulo de inicialização**, contacta os agentes para informar a troca de gerente e passa a se comportar como um gerente.

A execução do gerente começa com o **módulo de inicialização** que lê os arquivos de inicialização e carrega as estruturas de dados. Após, a execução é transferida a um **módulo de polling** que pesquisa uma estrutura chamada **lista de eventos** dos módulos de interação verificando a ocorrência de eventos. Cada módulo de interação possui uma *thread* de execução e sua função é inserir notificações sobre a ocorrência de eventos na respectiva lista de eventos. A implementação em lista impede que eventos sejam perdidos em caso de demora na interpretação.

Durante o funcionamento do gerente, eventos de energia são comunicados à **máquina de estados** para que esta programe as ações a serem executadas no tempo definido pelo administrador, em uma estrutura chamada **lista de ações**. Quando o momento de execução de uma ação confere com o tempo informado pelo **relógio**, esta é levada ao **interpretador de ações** para que seja enviada ao agente respectivo.

²⁷ Nos anexos A e B são apresentados exemplos de arquivos e configuração e de comportamento, respectivamente.

O relato de um evento de energia indica uma nova situação de energização da rede; portanto, as ações programadas pelo evento anterior não devem ser executadas. No caso de haver ações programadas por um evento para execução em momento futuro, e outro evento ser recebido, essas ações são canceladas e as ações relativas ao novo evento são programadas.

Ao receber uma ação a ser disparada, o **interpretador de ações** a transforma na seqüência desejada de ações mais simples, do tipo *get* e *set* e, quando necessário, traduz o estado desejado no estado suportado pelo equipamento. Após, as ações são enviadas ao módulo de **comunicação com agentes**.

As ações a serem disparadas são descritas no arquivo de comportamento. Antes de serem enviadas aos agentes, precisam ser transformadas em ações mais simples. As ações definidas são mostradas na tabela 6.1 e explicadas a seguir.

A ação *execute* indica que o agente do *equipamento* deve executar o *script* ou programa descrito pela cadeia de caracteres *programa* contida no parâmetro 3, no instante descrito através do parâmetro *tempo*. Esse cadeia de caracteres deve conter o caminho e o nome do *script* ou programa. O texto contido no parâmetro *mensagem* deve ser enviado imediatamente.

TABELA 6.1 - Ações definidas

<i>Ação</i>	<i>Parâmetro 1</i>	<i>Parâmetro 2</i>	<i>Parâmetro 3</i>	<i>Parâmetro 4</i>	<i>Parâmetro 5</i>
<i>execute</i>	<i>equipamento</i>	<i>tempo</i>	<i>programa</i>	<i>mensagem</i>	
<i>message</i>	<i>equipamento</i>	<i>tempo</i>	<i>mensagem</i>		
<i>shutdown</i>	<i>equipamento</i>	<i>flag</i>	<i>tempo</i>		
<i>wakeup</i>	<i>equipamento</i>	<i>tempo</i>	<i>mensagem</i>		
<i>state</i>	<i>equipamento</i>	<i>dispositivo</i>	<i>estado</i>	<i>tempo</i>	<i>mensagem</i>

A ação *message* indica que o agente de *equipamento* deve mostrar *mensagem* em *tempo*.

A ação *shutdown* indica que *equipamento* deve ser desligado em *tempo*. O texto contido no parâmetro *mensagem* deve ser mostrado ao usuário imediatamente. O parâmetro *flag* indica como o desligamento deve ser realizado. Os possíveis valores para *flag* podem ser:

- *savecontext* – o equipamento deve ter suas aplicações encerradas e o contexto deve ser salvo;
- *force* – o equipamento deve ter as aplicações encerradas mesmo que o contexto não possa ser salvo;
- *moretime* – o equipamento pode continuar operando se estiver realizando trabalho.

A ação *wakeup* é utilizada para ligar um equipamento. Como obviamente o agente não estará sendo executado, o gerente deve utilizar técnicas próprias para despertar o equipamento. A técnica mais utilizada atualmente, chamada de *Magic Packet*

[AMD96], consiste no envio de um pacote com o endereço MAC (*Media Access Control Address*) do equipamento repetido 16 vezes.

A ação *state* indica que *equipamento* deve ter o *dispositivo* especificado colocado em *estado* definido pelo parâmetro *tempo*. O texto contido no parâmetro *mensagem* deve ser enviado imediatamente.

A tradução do parâmetro *estado* é feita pelo **interpretador de estados**, que possui uma tabela contendo os equipamentos, seus dispositivos e o mapeamento entre os estados que o gerente utiliza e o estado que o agente implementa. Para adicionar novos dispositivos ou novos estados, é necessário apenas atualizar esta tabela (que deve ser armazenada em arquivo) e implementar um agente adequado. Não é necessário modificar o gerente nem a MIB proposta (seção 5.11).

A estrutura de dados que contém a lista de ações a serem executadas é chamada de *lista de ações futuras*, a qual contém registros com os campos mostrados na figura 6.4, os quais são explicados em seguida.

<i>Owner</i>	<i>TimeToExec</i>	<i>Action</i>	<i>Target</i>	<i>Flag</i>	<i>Time</i>	<i>Message</i>
--------------	-------------------	---------------	---------------	-------------	-------------	----------------

FIGURA 6.4 - Lista de ações futuras

O campo *Owner* identifica o módulo que programou a ação, permitindo que se eliminem as ações indesejadas programadas anteriormente; *TimeToExec* indica o momento em que a ação deve ser executada - esse campo é preenchido quando a ação é programada com a soma do valor atual do relógio e o valor do campo *Time*; o campo *Action* indica a ação a ser executada; *Target* identifica o equipamento ou grupo para quem a ação deve ser enviada; *Flag* é utilizado para a escolha de características específicas da ação; *Time* identifica o prazo após o qual a ação deve ser executada; *Message* contém a mensagem a ser mostrada ao usuário.

Essa lista é ordenada, em ordem crescente, pelo tempo em que as ações devem ocorrer. A execução de ações é controlada pelos pulsos do relógio e, a cada pulso, o campo *TimeToExec* do registro dos elementos da lista é analisado: todos que possuem valor inferior ou igual ao do **relógio** são executados e eliminados da lista. O valor do relógio é armazenado em uma variável de 32 bits que tem seu valor incrementado com regularidade, o que permite aproximadamente 4,2 bilhões de pulsos antes de zerar o contador (4.294.967.296), que equivalem a aproximadamente 138 anos, se a unidade de pulso for o segundo.

Além das ações que podem ser enviadas aos agentes, podem ser programados outros tipos de ações:

- confirmação de execução;
- envio de sinal de vida ao gerente;
- troca de modo de operação.

Ao enviar uma ação a um agente, o gerente deve programar uma ação futura de confirmação de execução. Ao receber uma interrupção de algum agente, o gerente deve verificar, na lista de ações futuras, se existe uma ação de confirmação de execução

correspondente e retirá-la da lista. Se o tempo de execução dessa ação for atingido, é registrada uma ocorrência no arquivo de *log*.

O funcionamento do reserva possui cinco estágios quando monitora o gerente, de acordo com a figura 6.5. Uma ação de envio de sinal de vida ao gerente é programada quando o reserva inicia a executar (1). Quando o relógio atingir o tempo programado, uma mensagem é enviada ao gerente, a ação é retirada da lista e são programadas uma ação de confirmação de execução e uma nova ação de envio de sinal de vida (2). Quando o gerente recebe a mensagem, deve responder ao reserva (3). Se a resposta não for recebida pelo reserva antes da ação de confirmação de execução ser disparada, o reserva envia outra mensagem ao gerente (4). Caso o gerente não responda também a este contato, o reserva assume a gerência (5).

Ações de troca de modo de operação permitem que o relógio envie eventos programados para a máquina de estados. Dessa maneira, a gerência de energia pode ser realizada baseada em horários e não apenas em eventos de energia. Esses eventos são interpretados da mesma maneira que eventos de energia, ou seja, a máquina de estado envia as ações correspondentes à lista de ações futuras.

Informações de *log* possibilitam o registro e a análise do comportamento da rede. As informações de *log* são armazenadas junto com uma estampa de tempo. Além destas informações, o gerente grava um registro de início de sessão e final de sessão toda vez que é inicializado ou finalizado. Assim, pode determinar se está iniciando uma nova sessão ou se recuperando de uma falha (no caso da ausência do registro de fim de sessão).

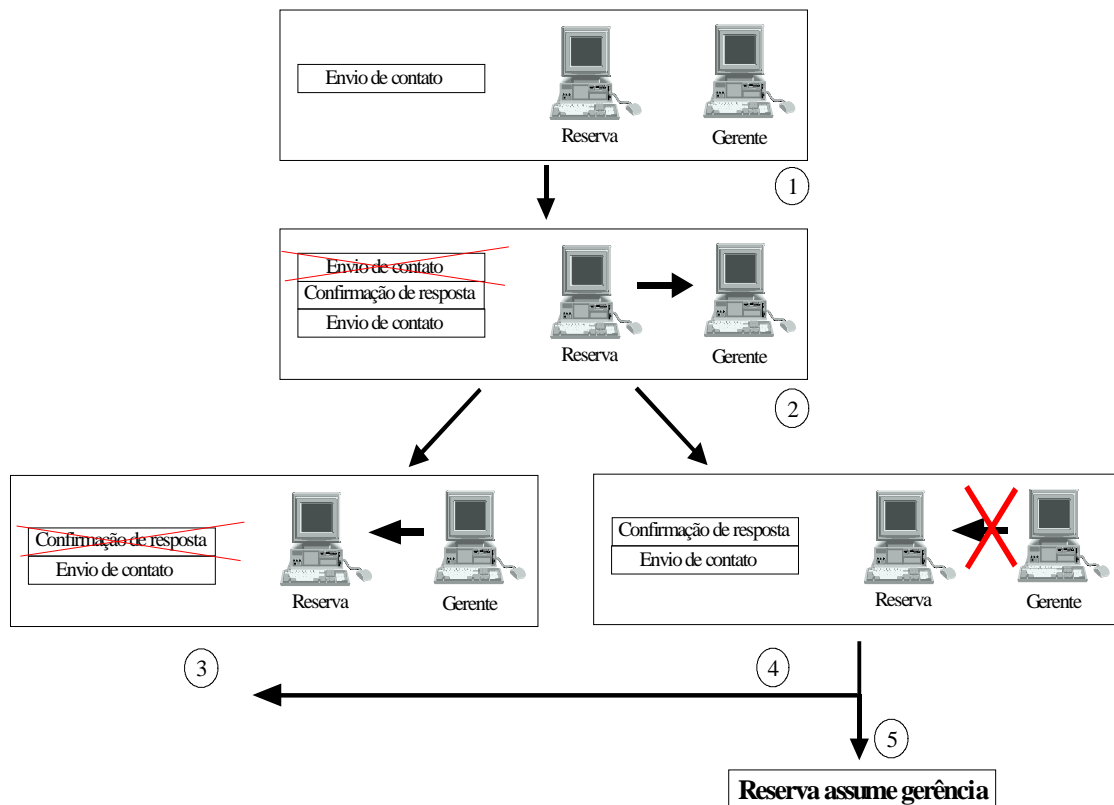


FIGURA 6.5 - Descrição de funcionamento do gerente reserva

São definidos níveis para a geração de *log* e o administrador pode escolher o nível desejado no arquivo de configuração. Um dos níveis serve para depuração (nível *debug*) e os outros dois servem para utilização durante funcionamento normal, um fornece informações detalhadas (nível *detail*) e outro fornece informações mínimas (*minimal*); o nível *none* não gera informações de *log*.

6.3 Configuração

O gerente foi projetado para ser adaptado a diversos ambientes computacionais, mas há aspectos que devem ser configurados. A configuração do gerente compreende a definição de suas próprias características (armazenadas em um **arquivo de configuração**) e de características do ambiente gerenciado (armazenadas em um **arquivo de comportamento**). A divisão em dois arquivos objetiva separar características que não mudam em função do ambiente de execução das características próprias de cada rede.

O administrador define as características através de uma interface e esta deve gerar os dois arquivos para posterior utilização pelo gerente, que pode funcionar sem interface a partir desse momento.

Os arquivos possuem sintaxe simples e são divididos em seções, conforme ilustrado na figura 6.6. Cada seção contém linhas formadas por dois campos: o nome de uma variável seguida por seu valor. Algumas seções podem ter sintaxe diferente, como a que descreve as ações a serem executadas (seção 6.3.2).

```
[seção]
    [subseção]
        variável=valor
        variável=valor
        ...
    [subseção]
        variável=valor
        variável=valor
        ...
[seção]
    ...
```

FIGURA 6.6 - Sintaxe dos arquivos

6.3.1 Arquivo de Configuração

O arquivo de configuração reúne as seções listadas a seguir na tabela 6.2. O anexo A contém um exemplo de arquivo de configuração.

TABELA 6.2 - Seções do arquivo de configuração

<i>Seção</i>	<i>Descrição</i>
<i>Miscellaneous</i>	Informações diversas do gerente
<i>States</i>	Tabela de tradução de estados das entidades gerenciadas

Seção *Miscellaneous*

Essa seção contém variáveis que descrevem diversos aspectos do gerente. Na tabela 6.3 essas variáveis são apresentadas e descritas.

TABELA 6.3 - Variáveis da seção *Miscellaneous*

<i>Variável</i>	<i>Descrição</i>
<i>events_port</i>	Porta a ser monitorada à espera de eventos de energia
<i>agents_port</i>	Porta que os agentes monitoram
<i>behavior_file</i>	Arquivo de comportamento a ser utilizado
<i>log_file</i>	Arquivo de <i>log</i> a ser utilizado
<i>log_level</i>	Nível de <i>log</i> desejado
<i>log_size</i>	Tamanho máximo do arquivo de <i>log</i>
<i>timeout</i>	Tempo de atraso permitido na resposta dos agentes
<i>retries</i>	Número de tentativas de comunicação com agentes antes de desistir
<i>clock</i>	Período do pulso de relógio em segundos

Seção *States*

Essa seção contém a tabela utilizada na tradução de estados. Essa tabela é montada de acordo com a figura 6.7. Um exemplo de seção *States* pode ser visto na figura 6.8. A definição completa pode ser vista no anexo A.

```
[states]
  [equipamento]
    [modelo/sistema operacional/padrão]
      [dispositivo]
        estado = estado
        estado = estado
        ...
      [dispositivo]
        ...
    [modelo/sistema operacional/padrão]
  ...
[equipamento]
  ...
...
```

FIGURA 6.7 - Seção de definição de estados

```

[states]
  [pc]
    [win98apm]
      [display]
        active           = active
        sleep_moretime = off_moretime
        sleep_context   = off_force
        sleep_force     = off_force
        off_moretime    = off_moretime
        off_context     = off_force
        off_force       = off_force
        reset           = reset
        logout          = off_force

```

FIGURA 6.8 - Exemplo de seção states

Essa seção é composta de subseções que traduzem os estados do sistema nos estados suportados pelos dispositivos. No exemplo da figura 6.8, como não há contexto a ser salvo em um monitor, o gerente é instruído a substituir os estados *sleep_context* e *off_context* por *sleep_force* e *off_force* respectivamente. Os estados existentes nos dispositivos são definidos por diversas características, como o tipo de equipamento (ou arquitetura), a identificação do modelo (ou do sistema operacional ou do padrão de gerência utilizado), e a natureza do dispositivo.

Essa estrutura permite que outros equipamentos, outros padrões de gerência e outros sistemas operacionais possam ser gerenciados pelo sistema sem necessidade de alteração do gerente.

6.3.2 Arquivo de Comportamento

Na tabela 6.4 são listadas as seções e sua respectiva descrição, constantes do arquivo de comportamento. No anexo B é apresentado um exemplo de arquivo de comportamento.

TABELA 6.4 - Seções do arquivo de comportamento

<i>Seção</i>	<i>Descrição</i>
<i>equipment</i>	Descrição dos equipamentos gerenciados
<i>groups</i>	Agrupamento dos equipamentos
<i>dependencies</i>	Relações de dependência entre equipamentos
<i>events</i>	Eventos que podem ser relatados pela UPS
<i>programmed_events</i>	Eventos disparados pelo relógio
<i>modes</i>	Relação entre os eventos e os modos de operação
<i>behavior</i>	Ações a serem executadas em cada modo de operação
<i>administrator</i>	Informações sobre o administrador

Para a descrição desse arquivo será utilizada uma rede fictícia, mostrada na figura 6.8. Nessa rede, o computador *Renoir* é o servidor de arquivos e também o

servidor HTTP. O gerente é executado nesse computador, que ainda possui interface de comunicação com a UPS; o *Hub* interliga os equipamentos da rede; o computador *Locatelli* é o servidor de impressão e a *impressora* possui uma chave eletrônica comandada por *Locatelli*, podendo ser ligada e desligada por *software*. A *cafeteira*²⁸ também é controlada por uma chave eletrônica através de *Rubens*; os demais computadores são estações de trabalho. Cabe ressaltar que nesta configuração não está previsto gerente reserva.

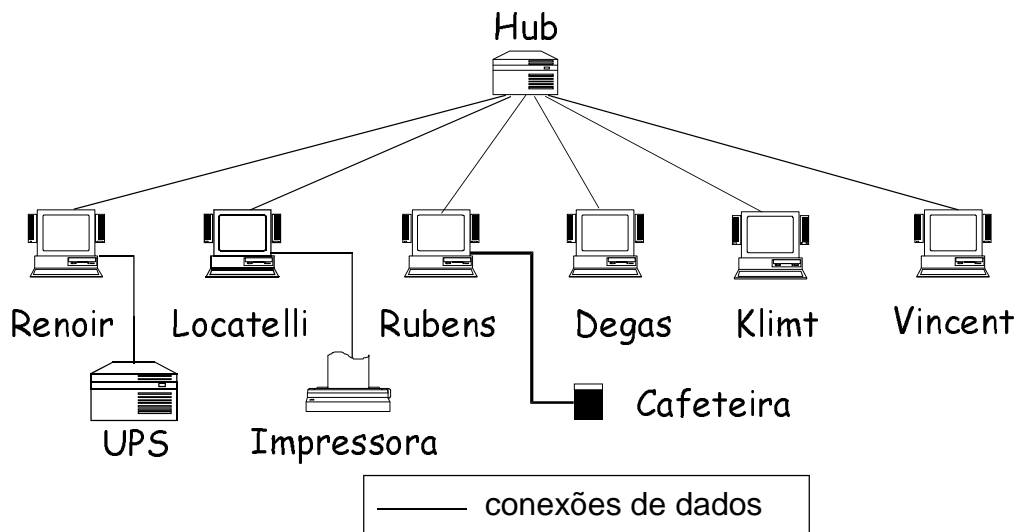


FIGURA 6.8 - Rede utilizada como exemplo

Seção Equipment

Descreve os equipamentos gerenciados através de um conjunto de variáveis listadas na tabela 6.5.

TABELA 6.5 - Variáveis da seção Equipment

<i>Variável</i>	<i>Descrição</i>
<i>name</i>	nome interno de identificação
<i>kind</i>	tipo de equipamento
<i>address</i>	endereço IP
<i>proxy</i>	endereço do agente <i>proxy</i>
<i>port</i>	porta do agente
<i>standard</i>	padrão de gerência
<i>std_version</i>	versão do padrão
<i>interactive</i>	indica se o equipamento é de uso interativo ou não
<i>more_time</i>	indica se o equipamento pode requisitar mais tempo caso seja informado de que deve ser desligado

²⁸ Cabe ressaltar que a cafeteira é usada simbolicamente para representar qualquer equipamento elétrico não essencial às atividades computacionais.

<i>Variável</i>	<i>Descrição</i>
<i>power_off</i>	indica se o agente pode desligar o equipamento
<i>power_on</i>	indica se o agente pode ligar o equipamento
<i>sleep_mode</i>	indica se o equipamento possui estado de espera
<i>device_control</i>	indica se o agente pode controlar os dispositivos
<i>dev_hd</i> , <i>dev_display</i> , <i>dev_processor</i>	indica se o agente pode atuar nesses dispositivos
<i>timeout</i>	tempo máximo de resposta
<i>retries</i>	número de tentativas antes de gerar alarme

Alguns itens dessa seção são obrigatórios para os equipamentos, como *name*, *kind*, *address* (ou *proxy*) e *standard*. Os outros são considerados opcionais e caso nenhum valor seja definido, o gerente assume que a característica não é suportada e utiliza comportamento padrão.

Seção Groups

Para simplificar o envio de ações aos agentes, os equipamentos podem ser agrupados de acordo com sua relevância para o funcionamento da rede e a preferência na utilização de energia. A abstração de grupos permite a utilização de *multicast* para o envio de ações. Um grupo pode conter subgrupos e equipamentos. A figura 6.9 propõe um possível agrupamento para os dispositivos do exemplo fictício.

Neste exemplo, comandos enviados para o grupo *equipamentos* serão difundidos a todos os equipamentos, enquanto que comandos enviados ao grupo *críticos* serão difundidos para *Hub* e *Renoir*. Se o administrador desejar, pode ainda enviar comandos individuais aos equipamentos. A difusão dos comandos acontece da raiz para as folhas na árvore.

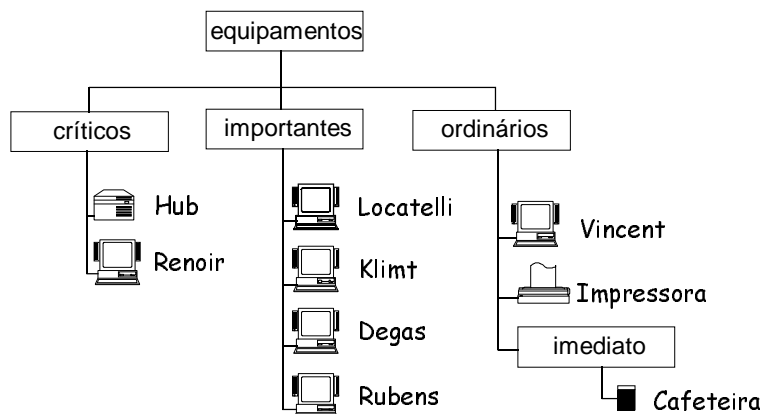


FIGURA 6.9 - Agrupamento de equipamentos

A sintaxe dessa seção é simples. Um grupo é definido na primeira vez em que aparece como variável e pode conter outros subgrupos ou equipamentos. A figura 6.10

mostra qual a sintaxe dessa seção e a figura 6.11, como essa seção é definida para o exemplo da figura 6.7.

```
[groups]
grupo = equipamento equipamento ...
grupo = equipamento grupo ...
...
```

FIGURA 6.10 - Seção de definição de grupos

```
[groups]
equipamentos= criticos importantes ordinarios
criticos     = hub renoir
importantes  = locatelli klimt degas rubens
ordinarios   = vincent impressora imediato
imediato     = cafeteira
```

FIGURA 6.11 - Exemplo de seção de definição de grupo

Seção Dependencies

A organização em grupos não é suficiente para representar todas as relações existentes entre os equipamentos. Por exemplo, de acordo com a figura 6.7, se o computador *Locatelli* for desligado, a *impressora* também pode ser desligada, já que depende de *Locatelli* para ser utilizada e gerenciada. Assume-se que o administrador esteja ciente das implicações que as definições dessas dependências terão sobre o funcionamento do sistema.

Relações de dependência podem ser construídas com a sintaxe descrita na figura 6.12. A figura 6.13 apresenta um exemplo de seção de relações de dependência para o exemplo da figura 6.7 - antes de o *Hub* ser desligado, os outros equipamentos devem ser desligados, pois não poderão se comunicar com a rede. Da mesma maneira, antes de *Degas* ser desligada, a *cafeteira* deve ser desligada.

```
[dependencies]
equipamento = equipamento equipamento ...
equipamento = equipamento grupo ...
...
```

FIGURA 6.12 - Seção de relações de dependência

```
[dependencies]
hub           = equipamentos
locatelli    = impressora
degas        = cafeteira
```

FIGURA 6.13 - Exemplo de seção de relações de dependência

Seção Events

Essa seção declara os eventos de energia que podem alterar o modo de operação da rede. Uma UPS pode informar diversos eventos mas, através dessa seção, o administrador pode escolher quais serão tratados pelo gerente. A figura 6.14 apresenta a sintaxe dessa seção; a figura 6.15 apresenta um exemplo de declaração de eventos.

```
[events]
events = evento evento ...
```

FIGURA 6.14 - Sintaxe da seção de declaração de eventos

```
[events]
Events = eUPSONBat eUPSLOWBat eUPSCritBat eUPSNormal eProgOff
```

FIGURA 6.15 - Declaração de eventos

Seção Programmed_events

Nessa seção os eventos programados são definidos. A tabela 6.6 mostra alguns exemplos de programação de eventos.

TABELA 6.6 - Programação de eventos

<i>Evento</i>	<i>Data</i>	<i>Hora</i>	<i>Modo</i>
<i>eProgOff</i>	13/03/2000	1:36	<i>_anymode</i>
<i>eProgOff</i>	<i>_everyday</i>	22:00	<i>m_normal</i>
<i>eProgOff</i>	<i>_everyday</i>	<i>_7200</i>	<i>m_normal</i>
<i>eProgOff</i>	<i>_once</i>	22:00	<i>_anymode</i>
<i>eProgOff</i>	<i>_sunday</i>	22:00	<i>_anymode</i>
<i>eProgOff</i>	<i>*/04/2000</i>	22:00	<i>_anymode</i>
<i>eProgOff</i>	<i>*/*2000</i>	22:00	<i>_anymode</i>
<i>eProgOff</i>	<i>10/*/*</i>	22:00	<i>_anymode</i>

O significado dos campos da tabela 6.6 é o seguinte: *evento* será enviado à máquina de estados em *data* e *hora* se o sistema estiver em *modo*.

Algumas palavras possuem significados especiais. Essas devem começar com o caracter *underscore* (*_*). No campo **Data**, as palavras *_sunday*, *_monday*, *_tuesday*, *_wednesday*, *_thursday*, *_friday* e *_saturday* representam os dias da semana. A palavra *_once* significa que o evento deve ser enviado apenas uma vez, quando o horário for atingido e a palavra *_everyday* indica que o evento deve ser enviado todos os dias no horário determinado. O asterisco é utilizado como uma máscara, significando qualquer dia, qualquer mês ou qualquer ano, dependendo da posição em que aparece.

No campo **Hora**, o caracter *underscore* indica que, ao invés de ser enviado em determinado horário, o evento deve ser enviado certo número de segundos após o sistema entrar em *modo*.

No campo **Modo**, a palavra *_anymode* significa que o evento deve ser enviado a partir de qualquer modo.

Seção Modes

Essa seção define quais os modos de operação do sistema e como os eventos se relacionam com eles. A figura 6.16 mostra a sintaxe da seção de definição de modos; a figura 6.17 mostra um exemplo de seção de declaração de modos.

```
[modes]
    evento = modo
    evento = modo
    ...
```

FIGURA 6.16 - Sintaxe da seção de declaração de modos

```
[modes]
    eUPSONBat      = m_onBat
    eUPSLOWBat     = m_lowBat
    eUPSCritBat    = m_critBat
    eUPSNormal     = m_normal
    eProgOff       = m_prog
```

FIGURA 6.17 - Exemplo de seção de declaração de modos

Seção Behavior

Nessa seção, o comportamento do sistema é descrito em função da alteração no modo de operação. A sintaxe da seção indica o modo e a seqüência de ações a ser executada na transição, conforme a figura 6.18; o anexo B contém um exemplo da seção de definição de comportamento.

```
[behavior]
    [modo]
        ação quem flag tempo mensagem
        ação quem flag tempo mensagem
        ...
    [modo]
        ...
    ...
```

FIGURA 6.18 - Sintaxe da seção de definição de comportamento

As ações que podem ser programadas foram descritas na tabela 6.1.

Seção Administrator

Essa seção descreve como os alarmes devem ser informados. Possui dados para que o administrador possa ser alertado aonde estiver. A tabela 6.7 mostra as variáveis definidas para essa seção.

TABELA 6.7 - Variáveis da seção Administrator

<i>Variável</i>	<i>Descrição</i>
<i>Name</i>	nome
<i>Address</i>	endereço da estação de trabalho
<i>Email</i>	<i>email</i> para envio
<i>Phone</i>	telefone a ser contactado

As variáveis podem ser definidas mais de uma vez, de maneira que diversos administradores podem ser cadastrados (os alarmes são enviados a todos).

7 Implementação

Apenas alguns módulos do sistema foram implementados com o objetivo de validar a proposta. Os módulos implementados foram (de acordo com a figura 6.3):

- Gerente
 - inicialização
 - polling
 - monitor de UPS
 - relógio
 - eventos de energia
 - eventos de relógio
 - máquina de estados
 - interpretador de ações
 - interpretador de estados
 - interação com agentes
- Agente
 - MIB proposta
 - execução de programas ou *scripts*
 - recebimento de mensagens e envio para o usuário
 - desligamento de monitor
 - colocação do computador em estado de baixo consumo.
 - *logout* de usuário
 - reinicialização (*reset*) do computador
 - desligamento do computador

Os módulos restantes, relativos à interface e ao gerente reserva, não foram implementados por falta de tempo. Estes módulos devem ser implementados pelo grupo de pesquisa, no futuro.

7.1 Ambientes e Ferramentas Utilizados

O gerente não depende de características específicas de determinada plataforma, portanto pode ser desenvolvido em qualquer plataforma. Foi desenvolvido um gerente com os módulos acima para o ambiente GNU/Linux Conectiva [CON99]. O ambiente GNU/Linux foi escolhido porque dispõe de ferramentas gratuitas de desenvolvimento e porque o porte do código para outros ambientes Unix é simples.

Foi utilizada a linguagem Free-Pascal [FRE99] para a programação. Esta linguagem é semelhante à linguagem Object Pascal e o compilador utilizado possui versão tanto para Windows quanto para Unix.

Para o desenvolvimento dos módulos de comunicação com SNMP foi utilizada a biblioteca SNMP++ [HEW99]. O SNMP++ é uma biblioteca de objetos em C++ que facilita o desenvolvimento de aplicações com SNMP por utilizar orientação a objetos. Existe versão da biblioteca para Windows e Unix. Além de ser gratuita, esta biblioteca foi sugerida por consultas à lista de discussão (winsnmp-1@mailbag.intel.com) e ao grupo de notícias (comp.protocols.snmp). Em breve, deverá ser lançada versão desta biblioteca com suporte ao SNMPv3²⁹.

O agente foi desenvolvido para o ambiente Windows 98 porque este foi o primeiro sistema operacional a possuir *driver* ACPI. Os outros sistemas operacionais para os quais já foram anunciados *drivers* ACPI são o Windows 2000, lançado em Fevereiro de 2000, e o Linux [ACP99], cujo projeto encontra-se em fase inicial, conforme já relatado. No entanto, o protótipo também pode ser utilizado em outras versões do Windows, como Windows 95 e Windows NT, excetuando-se a funcionalidade de desligamento do computador.

O agente foi implementado em um computador com suporte a ACPI dotado de sistema operacional Windows 98, e para seu desenvolvimento foi utilizado o ambiente Visual C++ da Microsoft. A biblioteca SNMP++ também foi utilizada para o desenvolvimento do agente.

O agente desenvolvido é capaz de gerenciar uma rede de equipamentos com sistema operacional Windows, realizando:

- *logout* de usuários;
- reinicialização do sistema operacional;
- encerramento do sistema operacional com ou sem salvamento de contexto;
- desligamento dos equipamentos com *driver* ACPI e gabinete ATX;
- colocação de equipamentos em estados de consumo reduzidos;
- desligamento do monitor de equipamentos com *driver* APM.

7.2 Experimentos

Os experimentos desenvolvidos consistiram em determinar o comportamento de alguns equipamentos em função das condições de fornecimento de energia. A figura 7.1 mostra a composição desta diminuta rede de equipamentos.

Para emular o comportamento de uma UPS, foi utilizado um emulador desenvolvido anteriormente pelo grupo. Este emulador é um *software* que se comunica pela porta serial do computador através do protocolo SDP-3, utilizado pelas UPS da empresa CP Eletrônica S.A. [CPE99]. Assim, o emulador comporta-se como uma UPS

²⁹ Segundo informações obtidas da lista de discussão.

da empresa CP Eletrônica. O emulador foi executado em um computador com Windows NT. A utilização do emulador tornou possível a simulação da ocorrência de eventos sem afetar o funcionamento da UPS que alimenta a rede.

Utilizou-se um agente *proxy* para permitir o monitoramento de UPSs da empresa CP Eletrônica. Este agente também foi desenvolvido como trabalho anterior do grupo, visto que alguns modelos de UPS não possuem comunicação por SNMP, e implementa a MIB proposta no trabalho de Peres [PER00]. Essa MIB possui diversos objetos para o controle da qualidade de energia fornecida e o monitoramento do funcionamento da UPS. Este agente foi executado em um computador com Windows NT e foi implementado como uma extensão ao agente Microsoft.

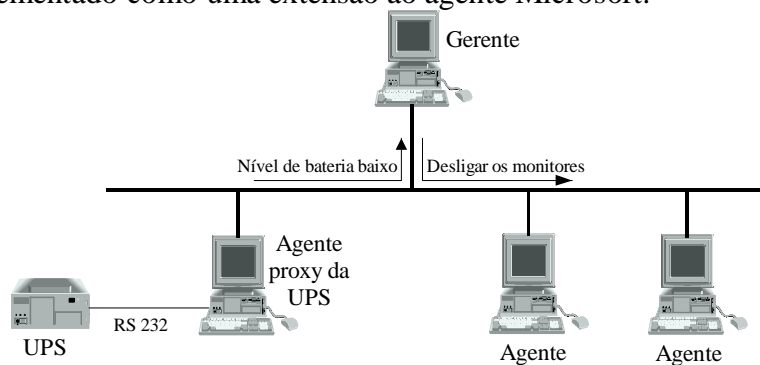


FIGURA 7.1 - Cenário do experimento

O gerente utiliza o protocolo SNMP para se comunicar com o agente *proxy* e obter o estado de fornecimento de energia. Para o experimento, foram monitorados quatro eventos: fornecimento normal, operação com baterias, nível de baterias baixo e nível de baterias crítico. Conforme destacado anteriormente, podem ser definidos outros eventos a serem analisados sem necessidade de alteração do código do gerente. Não existe método para a escolha dos eventos monitorado. A escolha desses quatro eventos foi feita pelo administrador da rede³⁰ em função da capacidade de informação da UPS e da estruturação dos equipamentos da rede.

O gerente foi executado em um computador com Linux. O agente de controle de consumo foi instalado em um computador com Windows 98. Este computador possui suporte a ACPI em sua placa mãe (modelo Intel SB440BX-2) e na fonte de alimentação (modelo ATX). Este computador também possui uma placa de rede (3Com 3C905C-TX-M) que permite que o equipamento seja despertado por *software*, quando for detectado um pacote de interesse na rede.

Para o experimento foi utilizado o cenário apresentado na figura 7.1, que representa parte dos equipamentos disponíveis no laboratório de pesquisa. Os eventos monitorados, os modos de operação da rede e os dispositivos controlados foram escolhidos de acordo com os equipamentos disponíveis e as necessidades da rede em questão. Os anexos A e B contêm os arquivos de configuração e comportamento utilizados nos experimentos.

Um experimento mais simples, que utilizou um emulador de eventos de linha de comando também foi realizado.

³⁰ Nesse caso, o autor do texto.

A figura 7.2 contém a tela da estação de gerência utilizada e a figura 7.3 contém a tela de uma estação gerenciada. A figura 7.2 contém três janelas. O gerente foi iniciado na janela situada no canto superior esquerdo. As informações de log do gerente foram direcionadas para a janela situada na região inferior da figura, e contém registro dos eventos reportados ao gerente e das ações enviadas aos equipamentos. Na janela do canto superior direito é executado um *software* emulador de UPS que apenas recebe uma *string* do console e a repete por *socket* para a porta em que o gerente monitora a UPS (consultar figuras 6.1 e 6.3).

```

[roger@vincent implementacao]$ saida/gerente

[roger@vincent roger]$ tty
/dev/pts/2
[roger@vincent roger]$ cd implementacao/
[roger@vincent implementacao]$ export LD_LIBRARY_PATH=.
[roger@vincent implementacao]$ saida/socketcli
Evento: eups_teste
Evento: eups_onbat
Evento: fim
[roger@vincent implementacao]$ saida/socketcli
Evento: eups_onbat
Evento: []

Início: Iniciando UpsThread...
Fim : Iniciando UpsThread...

M_ONBAT
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: REMBRAND em 0 s Mensagem: Equip. sera desligado quando a fonte exaurir Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: RUBENS em 0 s Mensagem: Equip. sera desligado quando a fonte exaurir Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: DEGRAS em 0 s Mensagem: Equip. sera desligado quando a fonte exaurir Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: KLINT em 0 s Mensagem: Equip. sera desligado quando a fonte exaurir Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: LOCATELLI em 0 s Mensagem: Equip. sera desligado quando a fonte exaurir Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: ADHARRA em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: VINCENT em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: REMBRAND em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: DEGRAS em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: KLINT em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: LOCATELLI em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: RENOIR em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: VINCENT em 0 s Mensagem: Equip. sera desligado imediatamente Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: ADHARRA em 0 s Mensagem: Equip. sera desligado imediatamente Resp.=0
Executando: Own: M_ONBAT Acao: MESSAGE Alvo: RUBENS em 0 s Mensagem: Rede operando com recursos limitados Resp.=0
Executando: Own: M_ONBAT Acao: SHUTDOWN Alvo: VINCENT em 0 s Mensagem: Equip. sendo desligado Resp.=0
Executando: Own: M_ONBAT Acao: SHUTDOWN Alvo: CAFETEIRA em 5 s Mensagem: Resp.=0
Executando: Own: M_ONBAT Acao: SHUTDOWN Alvo: ADHARRA em 5 s Mensagem: Resp.=0
Executando: Own: M_ONBAT Acao: SHUTDOWN Alvo: CAFETEIRA em 0 s Mensagem: Equip. sendo desligado Resp.=0
Executando: Own: M_ONBAT Acao: SHUTDOWN Alvo: IMPRESSORA em 0 s Mensagem: Equip. sendo desligado Resp.=0
Executando: Own: M_ONBAT Acao: SHUTDOWN Alvo: ADHARRA em 0 s Mensagem: Equip. sendo desligado Resp.=0

```

FIGURA 7.2 - Tela do gerente

A figura 7.3 contém a tela de uma das estações gerenciadas, onde aparecem duas janelas. A janela de baixo contém a mensagem enviada pelo gerente e a janela de cima contém dados para depuração do funcionamento do agente SNMP.

Neste exemplo, a notificação do evento *eUPSONBat* no emulador (janela superior direita da figura 7.2), originou o envio de uma seqüência de ações *message* e *shutdown* para os agentes (janela inferior da figura 7.2), de acordo com o arquivo de comportamento utilizado (o mesmo mostrado no anexo B).

Os experimentos não objetivaram coleta de dados. Estes experimentos foram realizados apenas para analisar a viabilidade de utilizar o modelo proposto no controle de consumo. Os experimentos demonstraram que o controle de consumo é viável com a tecnologia atualmente encontrada em computadores pessoais.

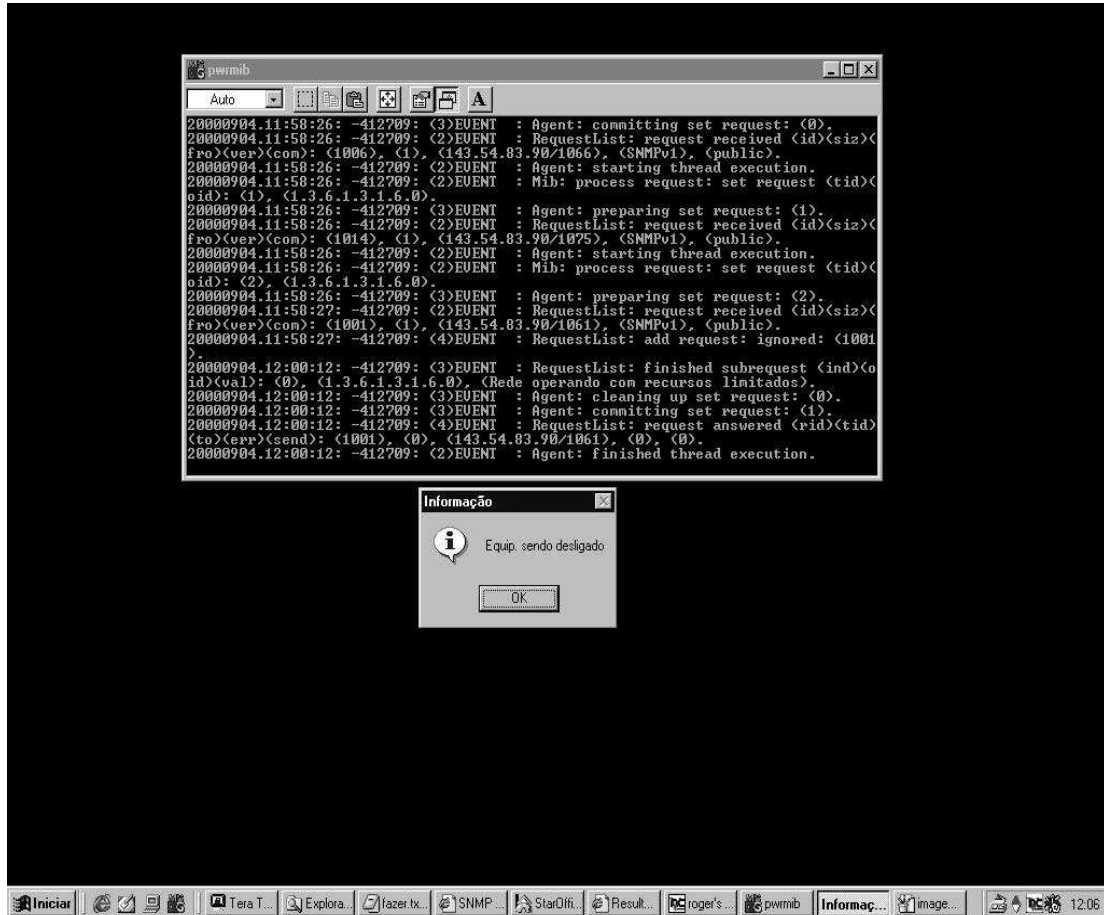


FIGURA 7.3 - Tela do agente

8 Conclusões e Perspectivas

Este capítulo apresenta uma análise dos resultados obtidos com a pesquisa e a implementação da solução proposta. Apresenta também uma relação de pesquisas futuras que devem ser realizadas.

8.1 Conclusões

Este trabalho propôs uma solução para o controle de consumo em redes de computadores em duas situações: durante operação, com recursos limitados de energia por falha no fornecimento convencional, e durante operação normal, com o objetivo de economizar recursos financeiros.

O desenvolvimento deste sistema de gerência envolveu as áreas de tolerância a falhas e gerência de redes. Os aspectos relativos à tolerância a falhas referem-se a:

- evitar que falhas no fornecimento de energia possam ocasionar perda de dados;
- aumentar o tempo de disponibilidade de carga das baterias para alimentar equipamentos de utilização crítica para a rede;
- evitar que falhas no gerente impeçam o controle de consumo;
- operar apesar de possíveis falhas nos agentes, fazendo com que elas se mantenham restritas às máquinas onde ocorrem.

Os aspectos de gerência de redes referem-se a:

- estudo do protocolo SNMP;
- estudo dos mecanismos de controle de consumo existentes nos computadores pessoais;
- desenvolvimento de *software* gerente;
- desenvolvimento de *software* agente.

A proposta de desenvolver este trabalho foi baseada na ausência de ferramentas similares no mercado e nos objetivos do projeto do grupo [JAN97]. Conforme relatado no trabalho, existem ferramentas para encerrar o sistema operacional em caso de falha no fornecimento de energia, mas até o presente momento não se conhece ferramentas que apresentem características com capacidade de interoperar com UPS de diferentes fabricantes, capacidade de monitorar diversas UPS e redes independentes, suporte a falhas do *software* de gerência e utilização de estados de consumo reduzido.

Foi implementado um protótipo para validação do sistema, que consistiu dos módulos essenciais do gerente e de um agente para controlar computadores com ACPI. Os resultados obtidos foram bastante promissores, indicando que a tecnologia atual

permite controle de consumo sem necessidade de adição de *hardware*, que possui custo elevado de implementação.

Dependendo da disponibilidade de mecanismos de controle de consumo nos equipamentos gerenciados é possível utilizar o sistema proposto para aumentar o tempo de duração das baterias da UPS em caso de falha no fornecimento comercial. Isto é feito através da execução de ações, tais como: desligar equipamentos de pouca importância para o funcionamento da rede desligar o monitor ou outros dispositivos dos equipamentos servidores encerrar o sistema operacional e salvar o contexto antes que a energia seja cortada colocar equipamentos e dispositivos internos em estado de consumo reduzido ou ainda ligar ou despertar equipamentos e dispositivos no momento em que são necessários, colocando-os em estado de consumo reduzido assim que possível.

Adicionalmente, como foi visto na motivação, o sistema também pode ser empregado para economizar recursos financeiros durante operação normal executando ações para: programar horários para o encerramento do sistema operacional e desligamento dos equipamentos como, por exemplo, à noite, em finais de semana, feriados ou períodos de férias; programar horários para a ligação automática de equipamentos; ligar ou despertar equipamentos no momento em que são necessários, colocando-os em estado de consumo reduzido assim que possível; e ainda complementar as políticas de controle de consumo locais, adaptando-as às necessidades e disponibilidades da rede.

Para chegar aos resultados finais, precisou-se estudar detalhadamente os mecanismos de controle de consumo existentes e o suporte para gerência de computadores que não fosse causar carga excessiva sobre o sistema.

Também foi preciso pensar em uma solução que, ao mesmo tempo, não efetuasse a troca de estado de energia das máquinas ao investigar em que condições de operação elas se encontravam, e não impedisse ao usuário a utilização remota das máquinas quando isto fosse do interesse dele - mesmo ocorrendo esta intenção no meio do final de semana, por exemplo!

As dificuldades na execução da solução foram principalmente devidas à falta de documentação referente a trabalhos relacionados, como estudos sobre a utilização de mecanismos de controle de consumo em ambientes computacionais, mencionando as vantagens e desvantagens deste controle, e estudos quantitativos sobre a economia de consumo conseguida, principalmente no contexto brasileiro. A única referência a estudos nesse âmbito foi encontrada no trabalho de Nordman [NOR96, NOR97] para o governo dos Estados Unidos da América.

A documentação sobre os padrões APM e ACPI também é escassa, tendo sido necessário fazer um estudo à parte destes padrões [KRO99]. Os *softwares* que são vendidos junto com as UPS possuem documentação apenas para uso do sistema.

Na implementação, a dificuldade encontrada decorreu do fato de que a tecnologia que permite controle total dos equipamentos é bastante recente e equipamentos e sistemas operacionais apenas passaram a ser disponibilizados no mercado durante o andamento do trabalho.

Deve-se salientar também que o tempo para o desenvolvimento do protótipo foi suficiente apenas para a implementação dos módulos principais do gerente e das principais funcionalidades do agente.

Além do desenvolvimento do sistema, esse trabalho gerou outros frutos, pois permitiu o estudo de diversas tecnologias, permitindo identificar suas vantagens e desvantagens, e possibilitou o acompanhamento de tendências que a informática está seguindo.

A própria proposta do sistema é uma contribuição interessante, pois permitiu que necessidades fossem identificadas e que soluções fossem apontadas, criando uma base inicial que trabalhos futuros possam utilizar. Além disso, foi desenvolvido um vocabulário próprio para a área e termos tiveram seu significado sedimentado.

Seria interessante que houvesse registro das primeiras reuniões com o grupo, quando as necessidades de um sistema de controle de consumo estavam sendo identificadas: havia uma grande quantidade de termos sem entendimento e tradução, havia dúvidas se o controle do *software* sobre o *hardware* era possível com a tecnologia existente, não se tinha idéia de como interagir com o *hardware* (sendo cogitada inclusive a construção de dispositivos a serem implantados dentro dos gabinetes dos computadores), não se sabia quais informações deviam ser utilizadas para a gerência. Naquela época o APM era o padrão utilizado, não se sabia que o Windows NT não possuía suporte à gerência de energia e o Windows 95 ainda era o sistema operacional largamente utilizado em estações de trabalho.

Em vista desse panorama inicial, a própria definição do escopo de atuação e a criação da proposta do sistema tiveram que ser realizadas um pouco às cegas, apresentando a solução prática e funcional que pode ser implantada em ambientes computacionais corriqueiros.

Por fim, cabe salientar que em todos os momentos em que foi necessário fazer uma definição sobre tecnologias e abordagens a serem utilizadas sempre procurou-se enfatizar a simplicidade e a utilidade prática. Seria frustrante desenvolver um sistema que não pudesse ser realmente aplicado por falta de suporte tecnológico ou por afastamento da realidade dos ambientes computacionais existentes.

8.2 Perspectivas Futuras

Existem diversas linhas de trabalho a serem seguidas após a conclusão deste. A perspectiva mais imediata é a implementação completa de todos os módulos do sistema. Após, devem ser realizadas medidas sobre a quantidade de energia economizada e estimada a sobrevida conseguida para a carga das baterias.

É necessário salientar que o padrão ACPI é relativamente novo e exige *hardware* específico para permitir controle pleno. Assim, uma solução construída hoje, sobre o parque computacional existente no Instituto de Informática ficaria principalmente restrita ao controle de monitores e desligamento de máquinas, sem permitir o controle individual de dispositivos.

Deve ser implementado um módulo de comunicação utilizando SNMPv3 tão logo que as ferramentas de desenvolvimento estejam disponíveis. O controle de ligação e desligamento remoto de equipamentos e dispositivos exige a existência de entidades autorizadas para tal fim, o que não pode ser conseguido com versões anteriores do SNMP.

Devem ser desenvolvidos agentes para outras plataformas computacionais de modo que o sistema possa ser utilizado em um ambiente heterogêneo. Por exemplo, no Instituto de Informática da UFRGS existem computadores com arquitetura PC, Apple Macintosh e Sun Microsystems. Os sistemas operacionais utilizados são Windows, Linux, Solaris e MacOS. Para o controle de todos esses equipamentos é necessário o desenvolvimento dos agentes específicos para cada plataforma.

Existem ainda equipamentos sem controle de consumo nativo, como impressoras e *hubs*. Para controlar esses equipamentos é necessário desenvolver, ou adquirir, *hardware* específico para controle de consumo.

Deve ser desenvolvida uma interface gráfica para acesso através da Web, de modo que o gerente possa ser controlado remotamente. O protocolo HTTP (*Hyper Text Transfer Protocolo*) e as linguagens HTML (*Hyper Text Markup Language*) e Java têm sido largamente utilizados para o desenvolvimento de interfaces. As vantagens dessa abordagem são:

- a inexistência de custo para desenvolver a interface com o usuário e treinar os usuários, já que o uso de navegadores é bastante difundido;
- a interoperabilidade entre as diversas plataformas em que a interface pode ser executada;
- a possibilidade de executar a interface de qualquer lugar da Internet.

Devem ser desenvolvidas versões para integração com plataformas de gerência. O uso de plataformas de gerência é bastante difundido atualmente. As plataformas de gerência oferecem um substrato de serviços que pode ser explorado objetivando uniformizar os procedimentos de gerência de maneira a facilitar o trabalho do administrador.

Devem ser adicionados mecanismos para controle de dispositivos com novas tecnologias de controle de consumo, como os barramentos IEEE 1394 e USB (*Universal Serial Bus*). As tecnologias de controle de consumo não param de se desenvolver. É necessário que novos mecanismos de controle de consumo sejam estudados para que o sistema seja mais abrangente.

Mecanismos de gerência distribuída devem ser estudados, pois podem aumentar a escalabilidade e a confiabilidade do sistema. Pode-se criar uma estrutura hierárquica de gerentes de controle de consumo para utilização em grandes instalações. Essa estrutura também poderia ser utilizada para mascarar falhas em algum gerente intermediário.

A tendência atual na pesquisa em gerência de redes é a utilização de outros protocolos em conjunto com o SNMP. Os mais utilizados são CORBA e Java RMI.

Devem ser estudadas possíveis vantagens destes protocolos, para que se possa definir sobre possíveis vantagens de uso.


```

        off_context = off_moretime
        off_force   = off_force
        reset       = reset
        logout      = off_moretime
[win2000apm]
  [system]
    active         = active
    sleep_moretime= sleep_moretime
    sleep_context=  sleep_context
    sleep_force    = sleep_force
    off_moretime   = off_moretime
    off_context    = off_context
    off_force      = off_force
    reset          = reset
    logout         = logout
  [display]
    active         = active
    sleep_moretime= sleep_moretime
    sleep_context=  sleep_force
    sleep_force    = sleep_force
    off_moretime   = off_moretime
    off_context    = off_force
    off_force      = off_force
    reset          = reset
    logout         = off_force
  [hd]
    active         = active
    sleep_moretime= sleep_moretime
    sleep_context=  sleep_context
    sleep_force    = sleep_force
    off_moretime   = off_moretime
    off_context    = off_context
    off_force      = off_force
    reset          = reset
    logout         = off_context
  [printer]
    active         = active
    sleep_moretime= off_moretime
    sleep_context=  off_moretime
    sleep_force    = off_force
    off_moretime   = off_moretime
    off_context    = off_moretime
    off_force      = off_force
    reset          = reset
    logout         = off_moretime
[win2000none]
  [system]
    active         = active
    sleep_moretime= off_moretime
    sleep_context=  off_context
    sleep_force    = off_force
    off_moretime   = off_moretime
    off_context    = off_context
    off_force      = off_force
    reset          = reset
    logout         = logout
  [display]
    active         = active
    sleep_moretime= off_moretime
    sleep_context=  off_force
    sleep_force    = off_force
    off_moretime   = off_moretime
    off_context    = off_force
    off_force      = off_force
    reset          = reset
    logout         = off_force
  [hd]
    active         = active
    sleep_moretime= off_moretime

```

```

sleep_context= off_context
sleep_force = off_force
off_moretime = off_moretime
off_context = off_context
off_force = off_force
reset = reset
logout = off_context
[printer]
active = active
sleep_moretime= off_moretime
sleep_context= off_moretime
sleep_force = off_force
off_moretime = off_moretime
off_context = off_moretime
off_force = off_force
reset = reset
logout = off_moretime
[winntacpi]
[system]
active = active
sleep_moretime= off_moretime
sleep_context= off_context
sleep_force = off_force
off_moretime = off_moretime
off_context = off_context
off_force = off_force
reset = reset
logout = logout
[display]
active = active
sleep_moretime= sleep_moretime
sleep_context= sleep_force
sleep_force = sleep_force
off_moretime = off_moretime
off_context = off_force
off_force = off_force
reset = reset
logout = off_force
[hd]
active = active
sleep_moretime= sleep_moretime
sleep_context= sleep_context
sleep_force = sleep_force
off_moretime = off_moretime
off_context = off_context
off_force = off_force
reset = reset
logout = off_context
[printer]
active = active
sleep_moretime= off_moretime
sleep_context= off_moretime
sleep_force = off_force
off_moretime = off_moretime
off_context = off_moretime
off_force = off_force
reset = reset
logout = off_moretime
[winntapm]
[system]
active = active
sleep_moretime= off_moretime
sleep_context= off_context
sleep_force = off_force
off_moretime = off_moretime
off_context = off_context
off_force = off_force
reset = reset
logout = logout

```

```

[display]
    active      = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_force
    sleep_force  = sleep_force
    off_moretime = off_moretime
    off_context  = off_force
    off_force    = off_force
    reset       = reset
    logout      = off_force
[hd]
    active      = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_context
    sleep_force  = sleep_force
    off_moretime = off_moretime
    off_context  = off_context
    off_force    = off_force
    reset       = reset
    logout      = off_context
[printer]
    active      = active
    sleep_moretime= off_moretime
    sleep_context= off_moretime
    sleep_force  = off_force
    off_moretime = off_moretime
    off_context  = off_moretime
    off_force    = off_force
    reset       = reset
    logout      = off_moretime
[winntnone]
[system]
    active      = active
    sleep_moretime= off_moretime
    sleep_context= off_context
    sleep_force  = off_force
    off_moretime = off_moretime
    off_context  = off_context
    off_force    = off_force
    reset       = reset
    logout      = logout
[display]
    active      = active
    sleep_moretime= off_moretime
    sleep_context= off_force
    sleep_force  = off_force
    off_moretime = off_moretime
    off_context  = off_force
    off_force    = off_force
    reset       = reset
    logout      = off_force
[hd]
    active      = active
    sleep_moretime= off_moretime
    sleep_context= off_context
    sleep_force  = off_force
    off_moretime = off_moretime
    off_context  = off_context
    off_force    = off_force
    reset       = reset
    logout      = off_context
[printer]
    active      = active
    sleep_moretime= off_moretime
    sleep_context= off_moretime
    sleep_force  = off_force
    off_moretime = off_moretime
    off_context  = off_moretime
    off_force    = off_force

```

```

        reset          = reset
        logout         = off_moretime
[win98acpi]
  [system]
    active            = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_context
    sleep_force      = sleep_force
    off_moretime     = off_moretime
    off_context      = off_context
    off_force        = off_force
    reset            = reset
    logout           = logout
  [display]
    active            = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_force
    sleep_force      = sleep_force
    off_moretime     = off_moretime
    off_context      = off_force
    off_force        = off_force
    reset            = reset
    logout           = off_force
  [hd]
    active            = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_context
    sleep_force      = sleep_force
    off_moretime     = off_moretime
    off_context      = off_context
    off_force        = off_force
    reset            = reset
    logout           = off_context
  [printer]
    active            = active
    sleep_moretime= off_moretime
    sleep_context= off_moretime
    sleep_force      = off_force
    off_moretime     = off_moretime
    off_context      = off_moretime
    off_force        = off_force
    reset            = reset
    logout           = off_moretime
[win98apm]
  [system]
    active            = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_context
    sleep_force      = sleep_force
    off_moretime     = off_moretime
    off_context      = off_context
    off_force        = off_force
    reset            = reset
    logout           = logout
  [display]
    active            = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_force
    sleep_force      = sleep_force
    off_moretime     = off_moretime
    off_context      = off_force
    off_force        = off_force
    reset            = reset
    logout           = off_force
  [hd]
    active            = active
    sleep_moretime= sleep_moretime
    sleep_context= sleep_context
    sleep_force      = sleep_force

```

```

        off_moretime = off_moretime
        off_context  = off_context
        off_force    = off_force
        reset        = reset
        logout       = off_context
[printer]
    active          = active
    sleep_moretime= off_moretime
    sleep_context=  off_moretime
    sleep_force    = off_force
    off_moretime  = off_moretime
    off_context   = off_moretime
    off_force     = off_force
    reset        = reset
    logout       = off_moretime
[win98none]
[system]
    active          = active
    sleep_moretime= off_moretime
    sleep_context=  off_context
    sleep_force    = off_force
    off_moretime  = off_moretime
    off_context   = off_context
    off_force     = off_force
    reset        = reset
    logout       = logout
[display]
    active          = active
    sleep_moretime= off_moretime
    sleep_context=  off_force
    sleep_force    = off_force
    off_moretime  = off_moretime
    off_context   = off_force
    off_force     = off_force
    reset        = reset
    logout       = off_force
[hd]
    active          = active
    sleep_moretime= off_moretime
    sleep_context=  off_context
    sleep_force    = off_force
    off_moretime  = off_moretime
    off_context   = off_context
    off_force     = off_force
    reset        = reset
    logout       = off_context
[printer]
    active          = active
    sleep_moretime= off_moretime
    sleep_context=  off_moretime
    sleep_force    = off_force
    off_moretime  = off_moretime
    off_context   = off_moretime
    off_force     = off_force
    reset        = reset
    logout       = off_moretime

```

Anexo B Exemplo de Arquivo de Comportamento

```
#                               Arquivo exemplo de comportamento
#
#                               criado em 22/09/99 por Roger Al-Alam Krolow (roger@inf.ufrgs.br)
#
#
#                               Regras de sintaxe:
#
#                               - suspenso (#) inicia comentário, até o final da linha
#                               - aspas (") delimitam texto lido como um único token
#                               - colchetes ([]) identificam início de seção
#                               - os caracteres: espaço ( ), vírgula (,), nova linha (carriage
#return) são separadores de tokens
#
#                               Regras de definição
#
#                               - o arquivo é dividido em seções identificadas pelo texto entre
#colchetes ([])
#                               - uma seção termina no início da próxima ou ao final do arquivo
#                               - as seções podem aparecer em qualquer ordem
#                               - seções com varias definições dos mesmos valores devem ter
#subseções (por exemplo, seção "equipment")
#                               - o arquivo é case insensitive
#
#
```

```
[equipment]
```

```
  [equipment1]
    name           = hub
    kind           = hub
    interactive    = no
    standard       = winntnone
    address        = 143.54.9.130
    more_time      = yes
    timeout        = 30
    retries        = 3
```

```
  [equipment2]
    name           = renoir
    kind           = pc
    interactive    = yes
    standard       = winntapm
    version        = 1.2
    address        = 143.54.9.130
    more_time      = yes
    timeout        = 30
    retries        = 3
```

```
  [equipment3]
    name           = locatelli
    kind           = pc
    interactive    = yes
    standard       = win98acpi
    version        = 1.0
#    software_power_off = yes
#    software_power_on  = yes
#    sleep_mode         = yes
#    device_control     = yes
#    system              = yes
#    display             = yes
#    hd                  = yes
```

```

#           processor          = yes
#           address            = 143.54.9.148
#           proxy               =
#           more_time          = no
#           timeout            = 30
#           retries            = 3

```

```

[equipment4]
  name          = rubens
  kind          = pc
  interactive   = yes
  standard     = winntapm
  version      = 1.2
  address      = 143.54.9.132
  more_time    = no
  timeout      = 30
  retries      = 3

```

```

[equipment5]
  name          = degas
  kind          = pc
  interactive   = yes
  standard     = winntapm
  version      = 1.2
  address      = 143.54.9.131
  more_time    = yes
  timeout      = 30
  retries      = 3

```

```

[equipment6]
  name          = klimt
  kind          = pc
  interactive   = yes
  standard     = winntapm
  version      = 1.2
  address      = 143.54.9.133
  more_time    = no
  timeout      = 30
  retries      = 3

```

```

[equipment7]
  name          = vincent
  kind          = pc
  interactive   = yes
  standard     = winntapm
  version      = 1.2
  address      = 143.54.9.134
  more_time    = no
  timeout      = 30
  retries      = 3

```

```

[equipment8]
  name          = impressora
  kind          = printer
  interactive   = no
  standard     = none
  software_power_off = yes
  software_power_on  = yes
  sleep_mode     = no
  device_control  = no
  proxy         = locatelli

```

```

        more_time      = no
        timeout        = 30
        retries        = 3

[equipment9]
    name              = cafeteira
    kind              = other
    interactive       = no
    standard          = none
    software_power_off = yes
    software_power_on  = yes
    sleep_mode        = no
    device_control     = no
    proxy             = degas
    more_time         = no
    timeout           = 30
    retries           = 3

[groups]
    equipamentos= criticos importantes ordinarios
    criticos    = hub renoir
    importantes = locatelli klimt degas rubens
    ordinarios  = vincent impressora imediato
    imediato    = cafeteira

[dependencies]
    hub          = equipamentos
    locatelli    = impressora
    degas        = cafeteira

[events]
    events = eUPSONBat eUPSLOWBat eUPSCritBat eUPSNormal eProgOff

[programmed_events]
    eProgOff    everyday    22:00    m_normal

[modes]
    eUPSONBat   = m_onBat
    eUPSLOWBat  = m_lowBat
    eUPSCritBat = m_critBat
    eUPSNormal  = m_normal
    eProgOff    = m_prog

[behavior]

[m_onBat]
    Message equipamentos "Rede operando com recursos limitados"
    Message importantes  "Equip. sera desligado quando a fonte exaurir"
    Message ordinarios    "Equip. sera desligado imediatamente"
    Shutdown ordinarios   MORETIME    0    "Equip. sendo desligado"
    Shutdown imediato     NOW          5    ""

[m_lowBat]
    Message equipamentos "O nivel de carga da fonte esta baixo"
    Message criticos      "Equip. sera desligado quando a fonte exaurir"
    Message importantes  "Equip. sera desligado em 20 segundos"
    ShutDown importantes MORETIME    20   "Equip. sendo desligado"
    ShutDown ordinarios  SAVECONTEXT 5    ""

[m_critBat]
    ShutDown equipamentos NOW          0    "Rede sendo desligada"

```



```
state      locatelli    display  off    0    ""

[m_normal]
Message    equipamentos "Fornecimento de energia restabelecido"
state      locatelli    display  on     0    ""

[m_Prog]
Message    ordinarios    "Iniciado desligamento programado"
ShutDown  ordinarios    MORETIME  5     ""

[administrator]

name       = "Roger"
address    = "locatelli"
email      = "roger@inf.ufrgs.br"
email      = "ingrid@inf.ufrgs.br"
phone      = "(51)99193300"
```

Bibliografia

- [ACP99] ACPI 4 LINUX (Home Page). Disponível em: <<http://phobos.fs.tum.de/acpi/index.html.en>>. Acesso em 14 dez. 1999.
- [AMD96] ADVANCED MICRO DEVICES. **Magic Packet Technology Application in Hardware and Software**. Disponível em: <<http://ftp.demax.sk/doc/demax/AMD/products/npd/techdocs/20381.pdf>>. Acesso em 28 ago. 2000.
- [ANE2000] ANEEL - Agência Nacional de Energia Elétrica. **Tarifas Médias por Classe de Consumo**. Disponível em <http://www.aneel.gov.br/sre/tarifas_medias.htm>. Acesso em 16 mar. 2000.
- [APC99] APC CORPORATION. **PowerChute Plus version 5.01 for Windows NT User's Guide**. 1999. Disponível em: <http://www.apc.com/tools/download/download.cfm?software_id=BROX%2D46ZRVF>. Acesso em 03 set. 1999.
- [APP99] APPLE COMPUTER INC. **Inside Macintosh: Processes**. 1999. Disponível em: <<http://developer.apple.com/techpubs/mac/Processes/Processes-2.html>>. Acesso em 11 nov. 1999.
- [BAY99] BAYTECH (Homepage). Disponível em: <<http://baytechdcd.com/welcome.shtml>>. Acesso em 08 jul. 1999.
- [BLA92] BLACK, U. **Network Management Standards**. New York. McGraw-Hill, 1992.
- [BLU99] BLUMENTHAL, U.; WIJNEN, B. **User Based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)**. RFC 2574. 1999. IETF. Disponível em: <<http://www.ietf.org>>. Acesso em 11 jun. 1999.
- [BUS99] BUSINESS SYSTEMS INTERNATIONAL. **BSI Remote Power Switch**. Disponível em: <<http://www.e-business.com/products/bsi-remote-power-switch.htm>>. Acesso em 22 jul. 1999.
- [CAS94] CASE, J. **UPS Management Information Base – Request For Comments 1628**. SNMP Research, Incorporated, maio 1994. Disponível em: <<http://www.atrete.ch/rfc-html/rfc/rfc1628.html>>. Acesso em 27 mar. 1998.
- [CON99] CONECTIVA S/A (Homepage). Disponível em: <<http://www.conectiva.com.br>>. Acesso em 14 dez. 1999.
- [CPE99] CP ELETRÔNICA S/A (Homepage). Disponível em: <<http://www.cp.com.br>>. Acesso em 14 dez. 1999.
- [DAT99] DATAPROBE (Homepage). Disponível em: <<http://www.dataprobe.com>>. Acesso em 22 jul. 1999.
- [DAV97] DAVID, B. **Group Communication as an Infrastructure for Distributed System and Network Management**. Jerusalem, Israel: Institute of Computer Science, The Hebrew University of Jerusalem, 1997. Dissertação de mestrado.

- [ENE98] ENERGY STAR (Homepage). Disponível em: <<http://www.epa.gov/energystar.html>>. Acesso em 12 jan. 1999.
- [ENE2000] ENERGY STAR. **Labeled Office Equipment**. Disponível em: <<http://www.epa.gov/appdstar/esoe/specpcmon.html>>. Acesso em 07 jan. 2000.
- [ENG99] ENGETRON S/A. **Manual do SUPSNet**. Disponível em: <<http://www.engetron.com.br/download.htm>>. Acesso em 14 dez. 1999.
- [FRE99] FREE PASCAL (Homepage). Disponível em <<http://www.freepascal.org>>. Acesso em 14 dez. 1999.
- [HEW97] HEWLETT-PACKARD. **HP OpenView Integration Series. Integration Concepts**. Fort-Collins: Hewlett-Packard Company, 1997.
- [HEW97a] HEWLETT-PACKARD. **HP OpenView Integration Concepts. OpenView Windows Developer's Guide**. Fort-Collins: Hewlett-Packard Company. 1997.
- [HEW99] HEWLETT PACKARD. **SNMP++ Home Page**. Disponível em: <<http://rosegarden.external.hp.com/snmp%2b%2b>>. Acesso em 24 dez. 1999.
- [IND99] INDUSTRIAL Computer Source. Disponível em: <<http://www.icsadvent.com/products/telecom/rpb-230.html>>. Acesso em 22 jul. 1999.
- [INT96] INTEL; MICROSOFT. **Advanced Power Management (APM) BIOS Interface Specification**. Revision 1.2. 1996. Disponível em: <<ftp://developer.intel.com/ial/powermgm/apmv12.pdf>>. Acesso em 10 nov. 1999.
- [INT96a] INTEL; MICROSOFT; TOSHIBA. *Advanced Configuration and Power Interface Specification*. 1996. Disponível em: <<http://www.teleport.com/~acpi/DOWNLOADS/ACPIsepc10.pdf>>. Acesso em 28 fev. 1999.
- [INT97] INTEL. **Instantly Available Power Managed Desktop PC**. Revision 1.1. 1997. Disponível em: <<ftp://download.intel.com/technology/IAPC/downloads/iapcdgr1.pdf>>. Acesso em 28 fev. 1999.
- [INT98] INTEL. **ATX Specification**. Versão 2.03. Disponível em: <http://www.teleport.com/~ffsupprt/spec/atx/atx2_03.pdf>. Acesso em 18 nov. 1999.
- [INT98a] INTEL. **Intel ATX Power Supply Design Guide**. Versão 0.9. Disponível em: <<http://www.teleport.com/~ffsupprt/spec/atx/atxps09.pdf>>. Acesso em 18 nov. 1999.
- [JAL94] JALOTE, P. **Fault Tolerance in Distributed Systems**. Englewood Cliffs: Prentice Hall, 1994.

- [JAN97] JANSCH-PÔRTO, I.; WEBER, T.; WEBER, R.; NETTO, J. **Sistema de Controle Confiável: Uma Aplicação de Energia**. Texto do projeto aprovado e parcialmente financiado pela FAPERGS. 1997. Processo nº 96/1561.7.
- [KRA97] KRAWCZYK, H. et al. **HMAC: Keyed-Hashing for Message Authentication**. RFC 2104. IETF. 1997. Disponível em: <<http://www.ietf.org/rfc/rfc2104.txt?number=2104>>. Acesso em 14 mar. 2000.
- [KRO99] KROLOW, R. **Gerência de Energia em Computadores Pessoais: trabalho individual**. Porto Alegre: PPGC da UFRGS, 1999.
- [KRO99a] KROLOW, R.; PERES, A.; JANSCH-PÔRTO, I. Management Information Base for Power Management in Computer Networks. In: IEEE LATIN AMERICAN NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, LANOMS, 1999, Rio de Janeiro. **Proceedings...** Florianópolis: UFSC, 1999.
- [LEB98] LEBOUTE, M.; SIQUEIRA, T.; JANSCH-PÔRTO, I. Gerenciamento Confiável de Energia em Redes de Computadores. In: WORKSHOP DE TOLERÂNCIA A FALHAS, 1., 1998, Porto Alegre. **Anais...** Porto Alegre: Instituto de Informática/UFRGS, 1998.
- [LIE99] LIEBERT CORPORATION. **SiteNet Multilink User Manual**. 1999. Disponível em: <http://www.liebert.com/products/english/products/software/engman/New_Manual/English/index.htm>. Acesso em 30 jun. 2000.
- [LOR97] LORCH, J.; SMITH, A. **Software Strategies for Portable Computer Energy Management**. Internal Report. University of California. Berkeley, CA: University of California, 1997. Disponível em: <<http://golem.cs.berkeley.edu/~lorch/papers/>>. Acesso em 28 fev. 1999.
- [MCQ97] MCQUINLAN, P. **Implementing Energy Star features on Network Computer Equipment**. [S.l.]:LANTech Network Specialists. E.E.C.A - Energy Efficiency and Conservation Authority, 1997.
- [MIC98] MICROSOFT CORPORATION. **Windows Power Management Configuration Tools Design Notes and Reference**. 1998. Disponível em: <<http://www.microsoft.com/hwdev/desinit/wpmttools.htm>>. Acesso em 28 fev. 1999.
- [MIC98a] MICROSOFT CORPORATION. **OnNow: Device Power Management**. 1998. Disponível em: <<http://www.microsoft.com/hwdev/desinit/DEVICEPM.HTM>>. Acesso em: 28 fev. 1999.
- [MIC98b] MICROSOFT. **Device Class Power Management Reference Specifications**. 1998. Disponível em: <<http://www.microsoft.com/hwdev/specs/PMref/PMrefSpec.htm>>. Acesso em 28 fev. 1999.
- [MIC99] MICROSOFT. **OnNow and Power Management**. Disponível em: <<http://www.microsoft.com/hwdev/onnow>>. Acesso em 27 jul. 1999.

- [MUL93] MULLENDER, S. **Distributed Systems**. 2nd. New York: Addison-Wesley, 1993.
- [NOR96] NORDMAN, B. et al. **Mesaured Energy Savings and Performance of Power-Managed Personal Computers and Monitors**. Berkeley, CA: Berkeley National Laboratory, 1996.
- [NOR97] NORDMAN, B. **User Guide to Power Management for PCs and Monitors**. University of California. 1997. Disponível por: <<http://eande.lbl.gov/EAP/BEA/LBLReports/39466/39466>>. Acesso em 12 jan. 1998.
- [PER98] PERES, A. **Análise de Tolerância a Falhas no protocolo SNMP**: trabalho individual. Porto Alegre: CPGCC da UFRGS, 1998.
- [PER00] PERES, A. **Definição e Construção de uma Base de Informações de Gerência para UPS**. Porto Alegre: PPGC da UFRGS, 2000. Dissertação de mestrado.
- [ROS91] ROSE, M. **The Simple Book - An Introduction To Management of TCP/IP-based Internets**. Upper Saddle River, NJ: Prentice-Hall. 1991.
- [SER99] SERVERTECHNOLOGY (Homepage). Disponível em: <<http://www2.servertech.com/products/ipms.html>>. Acesso em 22 jul. 1999.
- [STA96] STALLINGS, W. **SNMP, SNMPv2, and RMON - Practical Network Management**. 2nd. Reading Massachussets: Addison-Wesley, 1996.
- [STA98] STALLINGS, W. SNMPv3: A Security Enhancement for SNMP. **IEEE Communications Surveys**, New York, v. 1, n. 1. 1998. Disponível em: <<http://www.comsoc.org/pubs/surveys>>. Acesso em 29 ago. 2000.
- [STA98a] STALLINGS, W. Security Comes to SNMP: The new SNMP Proposed Internet Standards. **The Internet Protocol Journal**, [S.l.], v. 1, n. 3, 1998. Disponível em: <http://www.cisco.com/warp/public/759/ipj_issues.html>. Acesso em 29 ago. 2000.
- [WIJ99] WIJNEN, B. et al. **View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)**. RFC 2575. IETF. 1999. Disponível em: <<http://www.ietf.org/rfc>>. Acesso em 07 jan. 2000.