

# Gerenciamento Dinâmico de Energia para Interfaces de Rede

Bruna Maria Justino Cruz Vilar

Dissertação de Mestrado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Angelo Perkusich  
Orientador

Saulo Oliveira Dornellas Luiz  
Orientador

Campina Grande, Paraíba, Brasil

©Bruna Maria Justino Cruz Vilar, Julho de 2014

# Gerenciamento Dinâmico de Energia para Interfaces de Rede

Bruna Maria Justino Cruz Vilar

*Dissertação de Mestrado apresentada em Julho de 2014*

Angelo Perkusich

Orientador

Saulo Oliveira Dornellas Luiz

Orientador

Antonio Marcus Nogueira Lima

Componente da Banca

Jaidilson Jó da Silva

Componente da Banca

Campina Grande, Paraíba, Brasil, Julho de 2014

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

V697g Vilar, Bruna Maria Justino Cruz.  
Gerenciamento dinâmico de energia para interfaces de rede / Bruna Maria Justino Cruz Vilar. – Campina Grande, 2014.  
62 f. : il. color.

Dissertação (Mestrado em Ciências no Domínio da Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2014.

"Orientação: Prof. Dr. Angelo Perkusich, Prof. Dr. Saulo Oliveira Dornellas Luiz".  
Referências.

1. Gerenciamento Dinâmico de Energia. 2. Dispositivos Móveis. 3. Interfaces de Rede. I. Perkusich, Angelo. II. Luiz, Saulo Oliveira Dornellas. III. Título.

CDU 621.3:004.7(043)

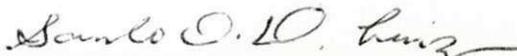
**"GERENCIAMENTO DINÂMICO DE ENERGIA PARA INTERFACES DE REDES"**

**BRUNA MARIA JUSTINO CRUZ VILAR**

**DISSERTAÇÃO APROVADA EM 29/07/2014**



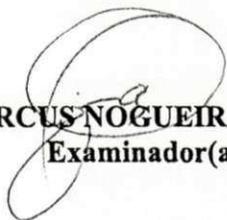
**ANGELO PERKUSICH, D.Sc., UFCG**  
**Orientador(a)**



**SAULO OLIVEIRA DORNELLAS LUIZ, D.Sc., UFCG**  
**Orientador(a)**



**JAÍLISON JOÃO DA SILVA, D.Sc., UFCG**  
**Examinador(a)**



**ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFCG**  
**Examinador(a)**

**CAMPINA GRANDE - PB**

# Agradecimentos

Agradeço a Deus e a Nossa Senhora das Graças, por todas as graças que recebo e sempre recebi, mesmo não sendo merecedora.

Agradeço aos meus pais, pelo amor, dedicação, paciência, cuidado e zelo.

Aos meus irmãos, cunhados e sobrinhos.

A Jocelma que me mostrou o caminho e não aceitou que eu fosse menos do que ela sabia que eu era capaz.

Ao CNPq pelo apoio financeiro.

A Diego Renato, pela disponibilidade em me ajudar e pela atenção de sempre.

Aos professores Saulo Dornellas de Oliveira Luiz e Angelo Perkusich pela oportunidade, orientação, respeito, e confiança em mim depositada.

A Pablo, que está comigo desde os tempos de escola, é meu amigo, meu amor e a minha maior fonte de inspiração.

# Resumo

Os avanços constantes nas tecnologias de *hardware*, *softwares*, têm tornado os sistemas computacionais móveis cada vez mais necessários no nosso dia a dia. O desenvolvimento de aplicativos cada vez mais complexos para esses sistemas computacionais, tais como os de edição de vídeos e jogos, exigem maior capacidade de processamento. O aumento do número e da complexidade desses aplicativos aliado à necessidade de conectividade com alguma interface de rede, implicam em maior consumo de energia, e assim menor autonomia da bateria.

Minimizar o consumo de energia é um problema fundamental no projeto de sistemas móveis, pois afeta diretamente a vida útil da bateria. Contudo, a densidade de energia armazenada nas baterias não tem aumentado na mesma taxa que o desempenho requerido por esses sistemas nos últimos anos, assim, a indústria de dispositivos móveis tem realizado esforços para estender a autonomia da bateria usando eficientemente a energia disponível e atendendo as demandas do usuário. Uma das técnicas bem sucedidas e bastante utilizada a nível de sistema é o *Gerenciamento Dinâmico de Energia* (GDE). A aplicação desta técnica reduz a dissipação de energia desligando ou reduzindo o desempenho de componentes do sistema, quando os mesmos encontram-se ociosos. No entanto, empregar essa técnica também acarreta em perda de desempenho devido a sobrecarga associada com desligamentos e religamentos de dispositivos. Uma política de gerenciamento eficiente deve, portanto, procurar maximizar a economia de energia, mantendo a penalidade do desempenho dentro de limites aceitáveis.

Palavras-chave: Gerenciamento Dinâmico de Energia, Dispositivos Móveis, Interfaces de Rede.

# Abstract

The constant advances in software and hardware technologies are making mobile computational systems increasingly needed in our daily activities. The development of increasingly complex applicative for these systems, like those for video editing and games, require higher processing power. The increase in number and complexity of these applicative together with the necessity of connectivity with some kind of network interface imply in higher power demand, and thus smaller battery autonomy.

Minimizing the consumed energy is a fundamental problem to the project of mobile systems, because that affects directly the life time of the battery. However, the amount of energy stored in the batteries is not increasing at the same rate as the performance required by these systems in the past few years, therefore, the industry of mobile devices has directed efforts to extend the autonomy of batteries by using more efficiently the available energy and paying attention to the demands of the user.

One of the most successful and used techniques at system level is Dynamic Power Management. The usage of this technique reduce the dissipation of energy turning off or reducing the performance of components of the system, when they are considered idle. However, the usage of this technique leads to loss in performance because of the overcharge caused by the shut down and wake up of the devices. One managing politic should aim to maximize the economy of energy, keeping the penalty of performance at acceptable levels. In recent years, mobile computing systems, such as laptops, smartphones and tablets, are consuming increasingly amounts of energy. The development of increasingly complex applicative for these systems, such as video editing and games, demand greater capacity of data processing. The increase in number and complexity of this applicative, together with the necessity of some kind of network interface, imply in a higher consume of energy, therefore, smaller battery autonomy.

Keywords: Dynamic Power Management, Mobile Computing Systems, Network Interface.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	2
1.2	Objetivos . . . . .	5
1.3	Metodologia . . . . .	5
1.4	Organização do Texto . . . . .	6
<b>2</b>	<b>Fundamentação Teórica e Revisão Bibliográfica</b>	<b>8</b>
2.1	Revisão Bibliográfica . . . . .	8
2.2	Políticas de Gerenciamento de Energia . . . . .	9
2.2.1	Políticas de Timeout . . . . .	9
2.2.2	Políticas Preditivas . . . . .	12
2.2.3	Políticas Estocásticas . . . . .	13
2.3	Sistemas Interativos e Cargas de Trabalho . . . . .	16
<b>3</b>	<b>Metodologia da Pesquisa</b>	<b>18</b>
3.1	Cargas de Trabalho . . . . .	18
3.2	Definição do Limiar de Ociosidade da Interface de Rede . . . . .	21
3.3	Políticas Utilizadas . . . . .	24
3.3.1	Política de Timeout . . . . .	24
3.3.2	Política Preditiva . . . . .	27
3.3.3	Política da Curva L . . . . .	28
3.3.4	Política Preditiva sem Religamento Automático . . . . .	29

3.4	Simulações . . . . .	30
3.4.1	Simulação da Política de Timeout . . . . .	30
3.4.2	Simulação da Política Preditiva . . . . .	32
3.4.3	Simulação da Política da Curva L . . . . .	37
3.4.4	Simulação da Política Preditiva sem Religamento Automático . . . . .	42
3.5	Curva de Pareto . . . . .	46
3.6	Simulação das Cargas de Trabalho Juntas . . . . .	47
<b>4</b>	<b>Resultados Experimentais</b>	<b>49</b>
4.1	Medição da Tensão e Corrente Fornecidas ao <i>Notebook</i> . . . . .	49
4.2	Estudo de Caso . . . . .	52
4.2.1	Especificações dos Estados de Energia de uma Interface de Rede . . . . .	52
4.2.2	Roteiro dos Experimentos . . . . .	55
4.3	Resultados Experimentais . . . . .	55
<b>5</b>	<b>Conclusões</b>	<b>57</b>
5.1	Trabalhos Futuros . . . . .	57
	<b>Referências Bibliográficas</b>	<b>59</b>

# Glossário

$T_{be}$	Break-Even Time
FS	Fila de Serviços
GDE	Gerenciamento Dinâmico de Energia
GE	Gerenciador de energia
PS	Provedor de Serviços
RS	Requisitante de Serviços

# Lista de Tabelas

4.1 Resultados dos experimentos. . . . .	56
--	----

# Lista de Figuras

1.1	Esquema de gerenciamento para carga de trabalho e dispositivo. . . . .	3
1.2	Ilustração das variações na potência média consumida para um sistema com GDE . . . . .	4
2.1	Ilustração do valor do $T_{be}$ necessário para compensar a energia adicional gerada na transição de estados. . . . .	9
2.2	Ilustração do valor da potência quando o intervalo ocioso é muito menor que o <i>timeout</i> selecionado . . . . .	10
2.3	Ilustração do valor da potência quando o intervalo ocioso é maior ou igual ao <i>timeout</i> selecionado. . . . .	11
2.4	Ilustração de períodos ociosos versus períodos ativos, denominada curva L.	13
2.5	Componentes do modelo de um sistema com gerenciamento de energia. . .	14
2.6	Exemplo em cadeia de Markov para um <i>requisitante de serviços</i> . . . . .	15
2.7	Ilustração da insatisfação do usuário. . . . .	17
3.1	Transmissão total ( <i>download</i> + <i>upload</i> ) para uma carga de trabalho Pessoal.	19
3.2	Transmissão total ( <i>download</i> + <i>upload</i> ) para uma carga de trabalho Escritório. . . . .	20
3.3	Transmissão total ( <i>download</i> + <i>upload</i> ) para uma carga de trabalho Jogo. .	20
3.4	Limiar baseado na transmissão total ( <i>download</i> + <i>upload</i> ) para uma carga de trabalho. . . . .	21
3.5	Função custo para a seleção do limiar de ociosidade da interface de rede. .	23
3.6	Curva dos períodos ociosos versus períodos ativos para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	23

3.7	Relação entre os valores de $Threshold_{ocioso}$ e $Threshold_{ativo}$ e a quantidade de acertos. . . . .	25
3.8	Diagrama de blocos para a simulação das políticas. . . . .	26
3.9	Curva da Potência média consumida versus $Timeout$ , para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	31
3.10	Política $Timeout$ : curva da Penalidade de desempenho versus $Timeout$ , para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	33
3.11	Política $Timeout$ : curva da Potência média versus Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	34
3.12	Política Preditiva: curva da potência média versus o parâmetro $a$ , para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	35
3.13	Política Preditiva: curva da Penalidade de desempenho versus o Parâmetro $a$ , para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	36
3.14	Política Preditiva: curva da Potência média versus a Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	38
3.15	Política Curva L: curva da Potência média versus o $Threshold$ para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	39
3.16	Política Curva L: curva da Penalidade de desempenho versus o $Threshold$ para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	40
3.17	Política Curva L: curva da Potência média versus Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	41
3.18	Política Preditiva SRA: curva da Potência média versus o Parâmetro $a$ para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	43
3.19	Política Preditiva SRA: curva da Penaliade de desempenho versus o Parâmetro $a$ para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	44
3.20	Política Preditiva SRA: curva da Potência média versus a Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo. . . . .	45
3.21	Curvas de Pareto para a carga de trabalho Pessoal. . . . .	46
3.22	Curvas de Pareto para a carga de trabalho Escritório. . . . .	47
3.23	Curvas de Pareto para a carga de trabalho Jogo. . . . .	47
3.24	Curvas de Pareto para a carga de trabalho Mista. . . . .	48

4.1	Diagrama de blocos para medição da tensão e corrente fornecidos ao <i>notebook</i> .	50
4.2	Diagrama elétrico para medição da tensão e corrente fornecidas ao <i>notebook</i> através do módulo de aquisição de dados NI USB-6210. . . . .	50
4.3	Diagrama de blocos no programa LabVIEW. . . . .	51
4.4	Painel frontal no programa LabVIEW. . . . .	51
4.5	Valores de tensão, corrente e potência da plataforma experimental. . . . .	52
4.6	Taxa total de transmissão de dados para um trecho da carga de trabalho Pessoal. . . . .	53
4.7	Simulação das Políticas para um trecho da Carga de Trabalho Pessoal . . .	53
4.8	Transições dos estados da interface de rede ligada/desligada e desligada/ligada, respectivamente. . . . .	54
4.9	Resultados das medições de potência para cada uma das políticas testadas.	56

# Capítulo 1

## Introdução

Os avanços constantes nas tecnologias de *hardware*, *softwares*, têm tornado os sistemas computacionais móveis cada vez mais necessários no nosso dia a dia [1]. O desenvolvimento de aplicativos cada vez mais complexos para esses sistemas computacionais, tais como os de edição de vídeos e jogos, exigem maior capacidade de processamento. O aumento do número e da complexidade desses aplicativos aliado à necessidade de conectividade com alguma interface de rede, implicam em maior consumo de energia, e assim menor autonomia da bateria [2].

Para suprir essa demanda de processamento são utilizados processadores mais modernos, com altas frequências de processamento e maior quantidade de núcleos, o que drena a bateria mais rapidamente que processadores com um único núcleo e operando em frequências mais baixas. [3].

Minimizar o consumo de energia é um problema fundamental no projeto de sistemas móveis, pois afeta diretamente a vida útil da bateria. Mesmo com os avanços conseguidos com tecnologias para baterias não tem sido possível acompanhar as necessidades crescentes por energia nos últimos anos [4,5]. Assim, a indústria de dispositivos móveis tem realizado esforços para estender a autonomia da bateria usando eficientemente a energia disponível e atendendo as demandas do usuário.

As técnicas de redução de consumo de energia são classificadas em estáticas e dinâmicas [6]. As técnicas estáticas são usadas na fase de projeto; são exemplos a síntese e a compilação para baixo consumo. Já as técnicas dinâmicas, denominadas *Gerenciamento Dinâmico de Energia* (GDE), possibilitam a redução do consumo de energia nesses sistemas em tempo de execução, através do desligamento ou redução da frequência/tensão de operação de componentes de sistemas que estão ociosos ou sub-utilizados. Segundo Lu, Benini e Micheli [7], gerenciamento dinâmico de energia é a escolha dinâmica de estados

de energia para um dado sistema com base na sua carga de trabalho. Num dispositivo, um *gerenciador de energia* (GE) monitora os estados dos componentes e do sistema e controla seus estados de energia através de um procedimento denominado *política de gerenciamento de energia* [5], que decide quais estados de operação vão assumir os componentes do sistema [8].

Uma das técnicas bem sucedidas e bastante utilizada no nível de sistema é o *Gerenciamento Dinâmico de Energia* (GDE), [9]. A aplicação dessa técnica proporciona redução no consumo de energia ao desligar ou reduzir o estado de consumo de componentes do sistema, quando os mesmos encontram-se ociosos.

Muitos recursos vêm sendo empregados na exploração de diferentes técnicas de gerenciamento dinâmico de energia. Segundo Dhiman e Rosing [10], as políticas de gerenciamento dinâmico existentes podem ser classificadas em políticas heurísticas e políticas estocásticas. Enquanto as políticas heurísticas são mais simples de serem implementadas elas não oferecem garantias sobre a relação economia de energia e penalidade de desempenho. Políticas estocásticas são mais complexas de se implementar mas oferecem garantia na relação economia de energia e a penalidade de desempenho. Entretanto, essa garantia é apenas para cargas de trabalho contantes.

Dentro da classificação de políticas heurísticas, encontram-se as políticas de *timeout* e preditivas. Essas políticas embora não apresentem garantias na relação entre penalidade de desempenho e redução no consumo de energia, podem ter seus parâmetros otimizados de forma a apresentar um desempenho que atenda às especificações de penalidade e economia de energia.

O emprego de qualquer técnica de gerenciamento dinâmico de energia acarreta em perda de desempenho devido à sobrecarga associada com desligamentos e religamentos de dispositivos. Uma política de gerenciamento eficiente deve, portanto, procurar maximizar a economia de energia, mantendo a penalidade do desempenho dentro de limites aceitáveis.

## 1.1 Motivação

Os avanços constantes nas tecnologias de *hardware*, *softwares*, têm tornado os sistemas computacionais móveis cada vez mais necessários no nosso dia a dia [1]. Estender a autonomia das baterias tornou-se naturalmente um grande desafio. Um esforço significativo vem sendo investido no desenvolvimento de baterias capazes de armazenar mais energia. Entretanto, os avanços nas tecnologias das baterias não crescem na mesma proporção que a demanda por energia nos dispositivos móveis [11]. Como alternativa, as técnicas

de gerenciamento dinâmico de energia alteram o estado de energia de componentes do sistema que estão ociosos, permitindo que o consumo de energia seja reduzido e com isso uma maior autonomia para as baterias [10].

Interfaces de rede estão presentes na maioria dos sistemas computacionais móveis [12], tais como *notebooks*, *ultrabooks*, *tablets* e *smartphones*. Aumentar a autonomia das baterias é cada vez mais necessário. É possível utilizar técnicas de gerenciamento de energia, de forma a alterar o estado de energia de interfaces de rede que estejam ociosas e com isso estender a autonomia das baterias.

O consumo de interfaces de rede ociosas é relevante. Neves et al. [13] apresentam um modelo de um sistema computacional. Para esse modelo, foi verificado que o dispositivo que apresentava maior consumo, quando ligado e ocioso, era o dispositivo de rede.

A premissa fundamental para a aplicabilidade de gerenciadores de energia é que os sistemas ou componentes do sistema experimentem cargas de trabalho não uniformes durante a operação normal. Cargas de trabalho não uniformes são comuns nas interfaces de rede na maioria dos sistemas interativos [8] e [14].

Gerenciamento Dinâmico de Energia (GDE) é uma técnica para a escolha dinâmica de estados de energia para um dado sistema com base na carga de trabalho [7].

Um Gerenciador de Energia (GE) monitora o sistema e os componentes e controla os estados de energia de cada componente do sistema. Esse procedimento é chamado de política de gerenciamento de energia [15], em que estados de energia são selecionados em função da carga de trabalho do sistema. Uma carga de trabalho consiste de requisições. Em discos rígidos, requisições são comandos de leitura e escrita; em interfaces de rede, são pacotes de dados enviados e recebidos. Quando ocorre uma requisição, o dispositivo é dito ocupado, caso contrário, o mesmo encontra-se ocioso [6].

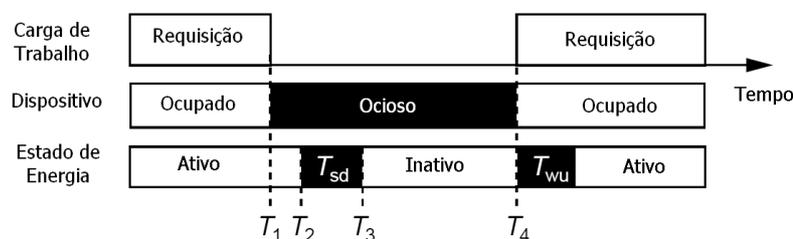


Figura 1.1: Esquema de gerenciamento para carga de trabalho e dispositivo.

Na Figura 1.1, é apresentado um esquema para o comportamento de uma carga de trabalho ao longo do tempo, sendo apresentadas duas requisições separadas por um período ocioso. O dispositivo encontra-se ocupado quando está atendendo uma requisição. Du-

rante o intervalo  $T_1$  e  $T_4$  não ocorrem requisições, e por isso o dispositivo está ocioso, é nesse intervalo que o dispositivo pode ser desligado para economizar energia. Em  $T_2$  o dispositivo é desligado e em  $T_4$  ocorre uma requisição e o dispositivo é ativado. Os intervalos de tempo  $T_{sd}$  e  $T_{wu}$  são os atrasos que o dispositivo leva para desligar e religar. Para muitos dispositivos, a mudança nos estados de energia proporciona uma sobrecarga de energia, o que torna o gerenciamento uma tarefa mais complicada, pois é preciso saber se a economia gerada pelo desligamento do dispositivo justifica a sobrecarga de energia e qual o impacto gerado pela penalidade de desempenho imposta ao religar o dispositivo.

Uma questão a ser considerada para as técnicas de gerenciamento de energia é a sua eficiência. Na Figura 1.2, é apresentado um esquema do comportamento de um sistema submetido à técnicas de gerenciamento diferentes. Em (a) é representado o comportamento de um sistema sem gerenciamento de energia, em (b) o comportamento com gerenciamento de energia ideal e em (c) é representado o comportamento do sistema para um gerenciamento não ideal.

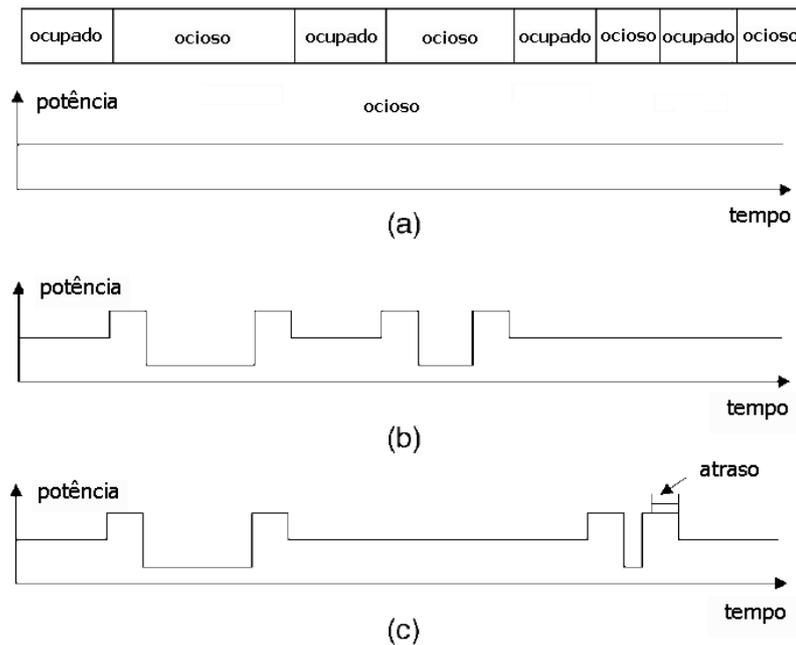


Figura 1.2: Ilustração das variações na potência média consumida para um sistema com GDE

Baseado nessa comparação é possível fazer algumas considerações:

1. se o gerenciador não identifica uma mudança de estado no dispositivo, ele deixa de agir e conseqüentemente deixa-se de economizar energia;
2. se o dispositivo permanece por muito tempo ocioso antes do gerenciador agir, também assim, deixa-se de economizar energia;

3. uma terceira situação ocorre, quando o gerenciador desliga um dispositivo ocioso mas não prevê com precisão o momento correto do religamento causando assim uma penalidade de desempenho.

Interfaces de rede, assim como outros dispositivos, quando não estão atendendo requisições da carga de trabalho, permanecem ligados. O consumo de uma interface de rede ociosa é significativo [16]. A meta de um gerenciador de energia é economizar energia e minimizar a penalidade de desempenho. Para este objetivo ser alcançado é preciso que as cargas de trabalho sejam conhecidas e que suas propriedades possam ser estudadas [5].

## 1.2 Objetivos

No contexto deste trabalho, o objetivo principal é comparar técnicas de gerenciamento dinâmico de energia para interfaces de rede, considerando a minimização do consumo de energia e o atendimento aos níveis de desempenho especificados.

Os objetivos específicos buscados foram:

- analisar abordagens para o gerenciamento dinâmico de energia com foco em interfaces de rede;
- caracterizar possíveis cargas de trabalho de interfaces de rede;
- realizar experimentos de medição de potência para comparar várias técnicas de gerenciamento dinâmico de energia.

## 1.3 Metodologia

A proposta dessa dissertação é comparar políticas para o gerenciamento dinâmico de energia em interfaces de rede;

Para caracterizar as cargas de trabalho para interfaces de rede, foi realizada a aquisição da taxa total de transmissão de dados (*download* + *upload*), gerada pelo uso de diferentes usuários, em diferentes sistemas computacionais. Para tanto, foi necessária a utilização de um *software* para realizar aquisição das taxas de *download* e *upload* enquanto as cargas de trabalho eram geradas. Para a caracterização dessas cargas apoiou-se na ideia utilizada por Luiz et al. [17] em que um limiar de ociosidade foi definido e a partir de comparações

pôde-se definir os intervalos da carga de trabalho que consistiam de períodos ociosos ou ativos.

Com as cargas de trabalho definidas foi possível traçar a curva L, e em seguida um mecanismo de classificação das cargas de trabalho foi desenvolvido.

Dando continuidade, várias políticas para o gerenciamento de energia foram estudadas, apresentadas no Cap 2. Nesse ponto, optou-se por investigar mais profundamente quatro políticas: *Timeout*, Preditiva, Curva L e Preditiva SRA. Essas políticas foram implementadas em um *software* interativo, onde pode ser comparado o desempenho das mesmas para as diferentes cargas de trabalho apresentadas nessa dissertação. Com isso, foi possível otimizar os parâmetros de cada uma das políticas de forma a obter valores de penalidade de desempenho e economia de energia otimizados.

Para concluir o que foi proposto inicialmente, as políticas foram implementadas em *software*, testadas em uma plataforma experimental, e através um módulo de aquisição de dados foi possível medir a tensão e corrente que eram fornecidas à plataforma. Os experimentos foram controlados, um trecho da carga de trabalho Pessoal foi escolhido, e antes da realização dos experimentos de medição de potência, todo o processo de simulação e otimização dos parâmetros foi repetido, dessa forma os resultados obtidos com as medições de potência puderam ser validados.

## 1.4 Organização do Texto

O texto está organizado em 4 capítulos, além desse capítulo introdutório, da seguinte forma:

- No Capítulo 2 é apresentado o conceito de gerenciamento dinâmico de energia e são mostrados alguns métodos para a aplicação de políticas de *timeout*, preditivas e estocásticas.
- No Capítulo 3 são caracterizadas algumas cargas de trabalho específicas para interfaces de rede. Também são apresentadas as técnicas de gerenciamento dinâmico de energia que foram utilizadas e são detalhadas as políticas que foram escolhidas para a realização desse trabalho. Em seguida são apresentados os resultados das simulações realizadas com as diferentes políticas para diferentes cargas de trabalho em interfaces de rede.
- No Capítulo 4 são apresentados os resultados experimentais desse trabalho.

- Por fim, no Capítulo 5 são apresentadas as conclusões e propostas de trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica e Revisão Bibliográfica

### 2.1 Revisão Bibliográfica

Os avanços constantes nas tecnologias de *hardware*, *softwares*, têm tornado os sistemas computacionais móveis cada vez mais necessários no nosso dia a dia [1]. Estender o tempo de descarga das baterias tornou-se naturalmente um grande desafio. Um esforço significativo vem sendo investido no desenvolvimento de baterias capazes de armazenar mais energia. Entretanto, os avanços nas tecnologias das baterias não cresce na mesma proporção que a demanda por energia nos dispositivos móveis [11]. Como alternativa, as técnicas de gerenciamento dinâmico de energia alteram o estado de energia de componentes do sistema que estão ociosos, permitindo que o consumo de energia seja reduzido e o tempo de descarga das baterias seja prolongado [10].

A ideia básica de um gerenciamento de energia é que a carga de trabalho do sistema não seja constante, verificando-se períodos ativos e ociosos, portanto, se é possível identificar o seu estado é possível reduzir o consumo de potência a partir de alguns mecanismos, ex.: desligamento do dispositivo que encontra-se ocioso. No entanto, apenas desligar o dispositivo não é suficiente. Quando um dispositivo é desligado e ocorre uma solicitação da carga de trabalho em um curto espaço de tempo é possível que a duração do período em que o dispositivo tenha ficado desligado não compense o consumo que é exigido para efetuar seu desligamento e religamento [18].

Um período inativo  $T_{idle}$  é definido como o período entre duas requisições consecutivas. Para um componente, de um modo geral, transições de estado geram consumo adicional de energia [19]. Quando  $T_{idle}$  é tempo suficiente para amortizar a energia adicional devido

à transição de estado, o componente pode ser colocado em estado de hibernação ou ser desligado para economizar energia. Assim, podemos definir um limiar, denominado *Break-Even Time*,  $T_{be}$ , que é o tempo ocioso mínimo necessário para atingir o ponto de equilíbrio entre o custo de transição e de economia de energia [20]. O valor do  $T_{be}$  pode ser obtido com a Equação (2.1). Uma ilustração do valor do  $T_{be}$  necessário para compensar a energia adicional gerada na transição de estados é apresentada na Figura 2.1. As constantes  $E_{sd}$  e  $E_{wu}$ , representam as energias para desligar e religar, respectivamente, um dispositivo. Os tempos  $T_{sd}$  e  $T_{wu}$  representam os tempos necessários para que um dispositivo desligue e religue, respectivamente. A constante  $P_w$  representa a potência média do dispositivo ligado, e  $P_s$  é a potência média do dispositivo desligado.

$$\begin{aligned}
 P_w t &\geq E_{sd} + E_{wu} + P_s(t - T_{sd} - T_{wu}) \\
 t &\geq \frac{E_{sd} + E_{wu} + P_s(T_{sd} + T_{wu})}{P_w - P_s} \\
 T_{be} &= \frac{E_{sd} + E_{wu} + P_s(T_{sd} + T_{wu})}{P_w - P_s}
 \end{aligned} \tag{2.1}$$

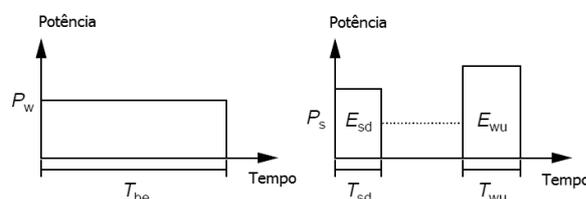


Figura 2.1: Ilustração do valor do  $T_{be}$  necessário para compensar a energia adicional gerada na transição de estados.

## 2.2 Políticas de Gerenciamento de Energia

Algumas das políticas básicas para gerenciamento de energia são: políticas de *timeout*, políticas preditivas e políticas estocásticas [6], [21]. Estas políticas têm sido amplamente utilizadas e serão explicadas a seguir.

### 2.2.1 Políticas de Timeout

Políticas de gerenciamento dinâmico de energia com base em *timeout* são amplamente aplicadas para dispositivos eletrônicos de consumo, tais como: o tempo de espera para desligar a tela ou o tempo para suspender todo o sistema. Uma política de *timeout*

estático é uma estratégia simples de gerenciamento de energia [1]. Quando um dispositivo permanece ocioso por um tempo maior que  $T_{be}$ , a política coloca o dispositivo em um estado de menor consumo de energia e o dispositivo volta à atividade quando ocorre uma requisição da carga de trabalho. Uma política de *timeout* adaptativo modifica o valor do *timeout* de forma a otimizar o seu desempenho. Existem evidências que mesmo uma política adaptativa simples [1] pode apresentar um melhor desempenho que uma política estática.

A política de *timeout*, [22], é a mais amplamente utilizada em muitas aplicações, tais como microprocessadores, monitores e discos rígidos, por causa de sua simplicidade. O valor do *timeout* pode ser fixado (*timeout* estático) ou pode ser alterado ao longo do tempo.

Vilar et al. [12] apresentaram uma comparação entre políticas de *timeout* dinâmico e *timeout* estático para interfaces de rede. A pesquisa realizada mostrou que a política de *timeout* dinâmico superava o de *timeout* fixo tanto na economia de energia quanto na penalidade de desempenho.

Para o caso de interfaces de rede, o objetivo de uma política de gerenciamento de energia com base em *timeout* é minimizar tanto o consumo de energia quanto a penalidade de desempenho causada quando a interface de rede está desligada mas o usuário queria usá-la. Nesse contexto, Luiz et al. [17] propuseram uma política dinâmica de *timeout* fixo para interfaces de rede, considerando a taxa total de transferência de dados (*download* e *upload*) e um *timeout*.

A potência média durante um intervalo ocioso  $t_i$  é função do *timeout* utilizado. Se  $t_i$  é menor que o *timeout*, como apresentado na Figura 2.2, o gerenciador de energia não enviará o comando para desabilitar a interface de rede. Dessa forma, a potência média consumida durante o intervalo ocioso será igual a potência correspondente ao estado ativo,  $P_{on}$ .

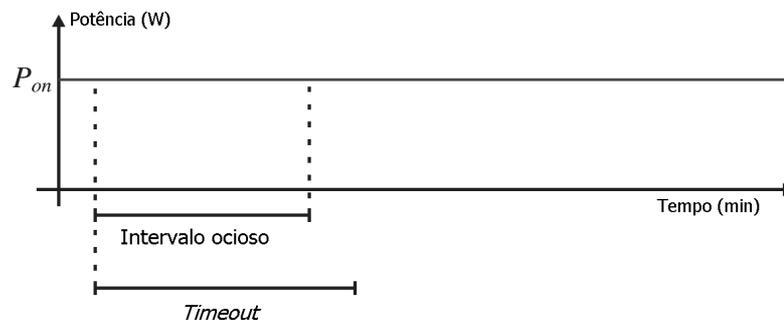


Figura 2.2: Ilustração do valor da potência quando o intervalo ocioso é muito menor que o *timeout* selecionado

Se o intervalo  $t_i$  é maior ou igual ao valor do *timeout*,  $\delta$ , selecionado, como apresentado na Figura 2.3, o gerenciador de energia envia o comando para desabilitar a interface de rede após o *timeout*. A variável  $T_s$  representa o intervalo de tempo que a interface leva para desligar,  $P_{on/off}$ , e religar,  $P_{off/on}$ . A energia gerada no processo de desligar e religar a interface é representado por  $E_s$ . A potência durante o período ocioso será:

- a potência da interface ligada durante o *timeout*  $P_{on}$
- a potência para desligar a interface  $P_{on/off}$  no tempo  $T_{on/off}$
- a potência durante o tempo em que a interface permanece desabilitada  $P_{off}$
- a potência para religar a interface  $P_{off/on}$  durante o intervalo de tempo  $T_{off/on}$ .

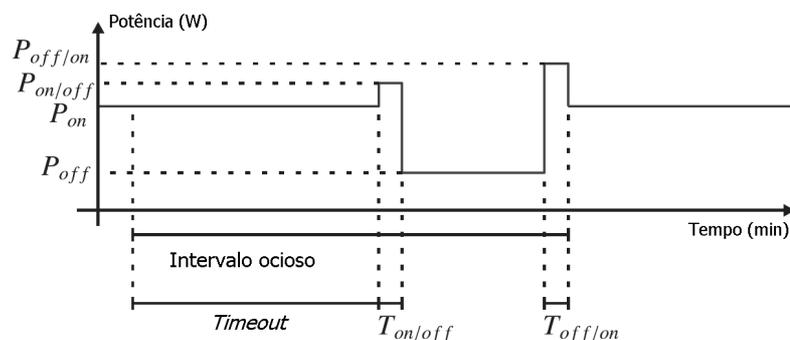


Figura 2.3: Ilustração do valor da potência quando o intervalo ocioso é maior ou igual ao *timeout* selecionado.

Portanto, considerando os casos mostrados nas Figuras 2.2 e 2.3 a potência média durante um intervalo ocioso é dada pela Equação (2.2) e é válida para  $T_s$  menor que o período de amostragem.

$$p_{med}(\delta, t_i) = \begin{cases} P_{on}, & \text{se } t_i < \delta \\ \frac{P_{on}\delta + E_s}{t_i + T_s}, & \text{se } t_i = \delta \\ \frac{P_{on}\delta + P_{off}(t_i - \delta - T_s) + E_s}{t_i}, & \text{se } t_i > \delta \end{cases} \quad (2.2)$$

A penalidade no desempenho também pode ser considerada. Pois se o usuário desejar utilizar a interface de rede e ela estiver desabilitada, o mesmo terá de esperar um tempo de transição  $T_{off/on}$  para poder utilizá-la. A probabilidade desse evento ocorrer é igual a  $Pr[T_i \geq \delta]$ .

## 2.2.2 Políticas Preditivas

Numa política preditiva, um período ocioso futuro é estimado com base nos períodos ociosos anteriores. O objetivo disso é que não seja necessário aguardar um período de *timeout* para desligar o dispositivo. Assim, em políticas preditivas, o período de espera pelo *timeout* é eliminado. Se é previsto que o dispositivo ficará ocioso por um tempo maior que o  $T_{be}$ , ele será desligado ao iniciar sua inatividade [23]. Na Figura 1.1 o tempo  $T_2 - T_1$  seria eliminado.

A questão crítica em políticas preditivas é prever com precisão qual a duração do próximo período ocioso. É com base nessa previsão que essa política atua, desligando ou não o dispositivo. No entanto, em sistemas em que a carga de trabalho não é conhecida a priori, como é o caso de sistemas computacionais de propósito geral, se tem pouco conhecimento a respeito das requisições futuras na carga de trabalho. Com isso, as políticas preditivas baseiam-se na premissa que o histórico de períodos ociosos passados irá se repetir em um futuro próximo, devido ao comportamento repetitivo de algumas aplicações [19].

Em [24], Lu propôs uma política que é uma combinação de *timeout* adaptativo e sistemas de previsão. Nessa política, as requisições que ocorrem no disco rígido são divididas em sessões. Um algoritmo prevê, a partir do comprimento das sessões anteriores, quando ocorrerá uma nova requisição. No período entre requisições, não há atividade no disco, e é esse o momento ideal para alterar o estado de energia do disco economizando energia e minimizando penalidades.

Outra política preditiva utilizada baseia-se na duração dos períodos ociosos passados para prever a duração dos períodos ociosos futuros, chamada de Árvore de Aprendizado Adaptativo. Essa política pode ser aplicada em sistemas ou componentes com um número arbitrário de estados de baixo consumo de energia, e com ela é possível prever qual o estado mais indicado para um dispositivo que inicia um período de inatividade. Segundo Chung et al. [25], essa técnica de gerenciamento dinâmico de energia foi usada em discos rígidos, e os resultados mostraram que a mesma é consistente e supera em termos de potência consumida e penalidade de desempenho de outras políticas preditivas e de *timeout*.

Hwang et al. [26] propuseram um esquema de média exponencial. A política estima a duração do próximo período ocioso pela média acumulada dos períodos ociosos e estimados anteriores. A fórmula de previsão recursiva é:

$$I_{n+1} = a.i_n + (1 - a).I_n \quad (2.3)$$

Na Equação (2.3),  $I_{n+1}$  é o novo valor previsto,  $I_n$  é o último valor previsto,  $i_n$  é

o último período ocioso verificado, e  $a \in (0, 1)$  é uma constante. Esse esquema propõe religamento preditivo para evitar atrasos de desempenho. A Equação (2.3) pode ser usada para prever o próximo período ocioso, que é função do último período ocioso e do valor previsto passado. O parâmetro  $a$  controla o peso relativo entre o histórico recente e passado na previsão. Se  $a = 0$ , então  $I_{n+1} = I_n$ , ou seja, a história recente não tem nenhum efeito sobre o valor previsto. Por outro lado, se  $a = 1$ , então  $I_{n+1} = i_n$ , e a previsão considera apenas o período ocioso mais recente.

Segundo Lu, Benini e Micheli [6], existe uma relação entre os períodos ociosos e ativos de uma carga de trabalho. Essa relação entre os períodos forma uma curva que possui a forma de  $L$ , como apresentado na Figura 2.4

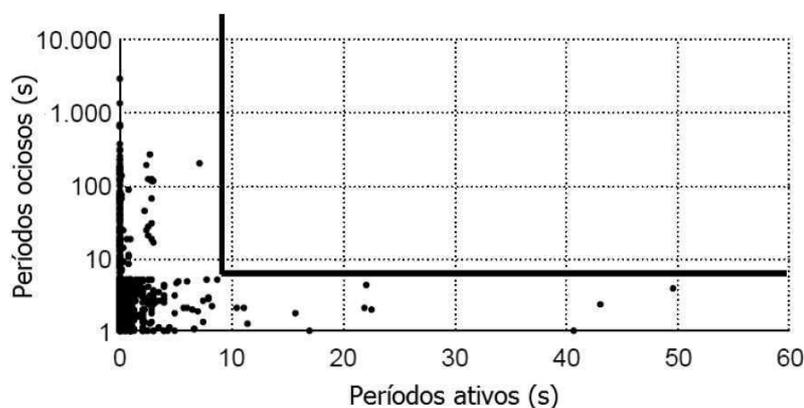


Figura 2.4: Ilustração de períodos ociosos versus períodos ativos, denominada curva  $L$ .

De acordo com Srivastava [14], se é verificado que para uma dada carga de trabalho a curva  $L$  é válida, é possível aplicar uma técnica de gerenciamento de energia baseada em um *threshold*. Se  $T_{ativo} < T_{thr}$  é assumido que o período ocioso subsequente será maior que o  $T_{be}$  e dessa forma o sistema pode ser desligado. A razão dessa política é que curtos períodos de atividade são frequentemente seguidos por longos períodos de ociosidade. Claramente, a escolha do  $T_{Thr}$  é difícil. Para isso, é necessária a análise cuidadosa da curva de períodos ociosos versus períodos ativos para que se possa chegar a um valor correto, dessa forma, a escolha de um  $T_{Thr}$  é um método *offline*, ou seja, baseado na extensiva coleta e análise de dados. Além disso, tal método não é aplicável a uma carga de trabalho cuja a curva dos períodos ativos e ociosos não possua a forma de um  $L$ .

### 2.2.3 Políticas Estocásticas

Políticas estocásticas modelam a chegada de requisições e mudanças nos estados dos dispositivos como processos estocásticos, tais como processos de Markov. Minimizar o

consumo de energia é um problema de otimização estocástico. Para um dado tempo, requisições chegam ao dispositivo com uma dada probabilidade e o estado do dispositivo é alterado com uma outra probabilidade. A probabilidade ótima para a mudança no estado do dispositivo pode ser obtida resolvendo um problema de otimização estocástico. Tal solução é válida apenas para distribuições geométricas em processos estocásticos estacionários. Há dois caminhos para generalizar esse modelo [6]: estendê-lo para outros modelos estocásticos e/ou incluir o comportamento não estacionário. A primeira extensão pode ser alcançada utilizando modelos de Markov contínuos no tempo. A segunda considera requisições não estacionárias. Um sistema com gerenciamento de energia é modelado por Benini et al. [8] como um *provedor de serviços* (PS), um *requisitante de serviços* (RS), uma *fila de serviços* (FS), e um *gerenciador de energia* (GE), como mostrado na Figura 2.5. O provedor de serviços processa as requisições de um único *requisitante de serviços* (RS), e possui vários estados de operação, como por exemplo: ativo, ocioso, desligado; cada estado é caracterizado por sua capacidade ou incapacidade de prover um serviço e um nível de consumo de energia. O *gerenciador de energia* (GE) é a unidade de controle que controla as transições de estados de energia do provedor de serviços. É possível que haja uma *fila de serviços* (FS) que armazena requisições que ainda não puderam ser atendidas pelo provedor de serviços, ou porque ele está ocupado ou está com taxa de serviço nula.

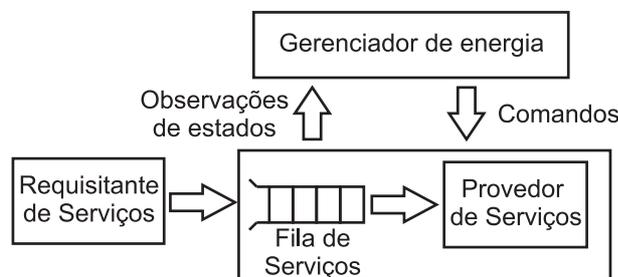


Figura 2.5: Componentes do modelo de um sistema com gerenciamento de energia.

Em uma política estocástica simples, os componentes do sistema com gerenciamento de energia, como apresentados na Figura 2.5, são modelados como cadeias de Markov estacionárias discretas no tempo. Assume-se que:

- os pedidos são indistinguíveis, não havendo prioridade de serviço;
- a fila de serviços possui tamanho finito.

O gerenciador de energia toma decisões com respeito ao provedor de serviços com base na história do provedor de serviços, no requisitante de serviço e na fila de serviço, e emite comandos para alterar o estado do provedor de serviços. O tempo é considerado discreto:

$t = 0, 1, 2, \dots$ . E o intervalo  $h$  entre dois instantes consecutivos de tempo é denominado *período de amostragem*.

A carga de trabalho do sistema é modelada como um requisitante de serviços como definido a seguir.

**Definição 1** *Um requisitante de serviços (RS) é um par  $(\mathcal{M}_{RS}, z(r))$  em que:*

- $\mathcal{M}_{RS}$  é uma cadeia de Markov estacionária com conjunto de estados  $\mathcal{R} = \{r_i | i = 0, 1, \dots, (R-1)\}$  e matriz de transição  $\mathbf{P}^{RS}$ ;
- $z(r)$  é uma função  $z : \mathcal{R} \rightarrow \mathbb{N}$  que representa o número de requisições lançadas por período pelo requisitante de serviços quando no estado  $r$ .

Um exemplo em cadeia de Markov para um *requisitante de serviços* (RS) com dois estados é apresentado na Figura 2.6.

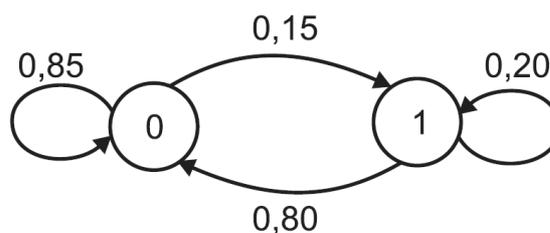


Figura 2.6: Exemplo em cadeia de Markov para um *requisitante de serviços*.

**Definição 2** *Um provedor de serviços (PS) é uma tripla  $(\mathcal{M}_{PS}(a), b(s,a), c(s,a))$  em que:*

- $\mathcal{M}_{PS}(a)$  é uma cadeia de Markov estacionária controlável com conjunto de estados  $\mathcal{S} = \{s_i | i = 1, 2, \dots, S\}$ , conjunto de comandos  $\mathcal{A} = \{a_i, | i = 1, 2, \dots, A\}$  e matriz de probabilidades de transição  $\mathbf{P}^{PS}(a)$ ;
- $b(s,a)$  é uma função  $b : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ , que representa uma taxa de serviço associada a cada estado  $s \in \mathcal{S}$  e comando  $a \in \mathcal{A}$ . Tal taxa de serviço é a probabilidade do provedor de serviços completar o atendimento de uma requisição de serviço num período, dado que o provedor de serviços está no estado  $s$  e comando  $a$  foi passado no início do período;
- $c(s,a)$  é uma função  $c : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ , que representa uma métrica de consumo de potência, corrente ou energia associada a cada estado  $s \in \mathcal{S}$  e comando  $a \in \mathcal{A}$ , i.e. o consumo (em unidades arbitrárias:  $W$ ,  $A$  ou  $J$ ) do PS num período, dado que o comando  $a$  foi tomado e o PS está no estado  $s$ .

**Definição 3** Uma fila de serviço (FS) de tamanho  $Q - 1$  é uma cadeia de Markov estacionária controlável  $\mathcal{M}_{FS}(a, s, r)$  com conjunto de estados  $\mathcal{Q} = \{q_i | i = 0, 1, \dots, (Q - 1)\}$ , variável de controle  $\theta = (a, s, r)$ , conjunto de comandos  $\Theta = \mathcal{A} \times \mathcal{S} \times \mathcal{R}$  e matriz de transição  $\mathbf{P}^{FS}(a, s, r)$ . As probabilidades de transição são definidas em (2.4).

$$p_{q_i, q_j}^{FS}(a, s, r) = \begin{cases} 1 - b(s, a), & \text{se } j = i + z(r) \text{ e } 0 < i + z(r) < Q \\ b(s, a), & \text{se } j = i + z(r) - 1 \text{ e } 0 \leq i + z(r) - 1 < Q \\ 1, & \text{se } j = i = 0 \text{ e } z(r) = 0 \\ 1, & \text{se } j = Q - 1 \text{ e } i + z(r) \geq Q \\ 0, & \text{caso contrário} \end{cases} \quad (2.4)$$

A partir dos modelos de cadeia de Markov para o provedor, a fila, e o requisitante de serviços, é possível formular um problema de otimização de política que consiste na escolha da política que minimiza o consumo de energia esperado do *provedor de serviços* para uma da restrição de penalidade e desempenho.

## 2.3 Sistemas Interativos e Cargas de Trabalho

Muitas políticas focam na economia de energia colocando em segundo plano o impacto no desempenho do sistema. Entretanto, em sistemas interativos, considerar o impacto no desempenho é prioritário e constitui uma tarefa difícil quantificar o incômodo do usuário, bem como prever as cargas de trabalho [6].

A análise do desempenho de um sistema em seu pior caso gera atrasos no atendimento às requisições da carga de trabalho. Durante a edição de um texto, por exemplo, o usuário poderia fazer perguntas do tipo “Quantas vezes eu terei que esperar durante a edição desse arquivo?”. Muitos usuários compreendem que alguns atrasos são inevitáveis, e quando ocorrem com baixa frequência podem ser relevados ou tolerados. No entanto, se atrasos ocorrem frequentemente, a insatisfação do usuário cresce drasticamente. Yung e Giovanni [6] chamam esse fenômeno de “efeito de memória curta”, pois usuários esquecem atrasos que ocorreram a muito tempo atrás. No entanto, se o intervalo de tempo entre os atrasos é demasiadamente curto, o usuário irá lembrar de atrasos consecutivos em um pequeno intervalo de tempo e seu sentimento de insatisfação tenderá a crescer. Consequentemente, se o tempo entre os atrasos é demasiadamente grande, o usuário tende a aceitar melhor o segundo atraso. Essa relação entre o intervalo de tempo entre os atrasos e a insatisfação do usuário é ilustrada na Figura 2.7.

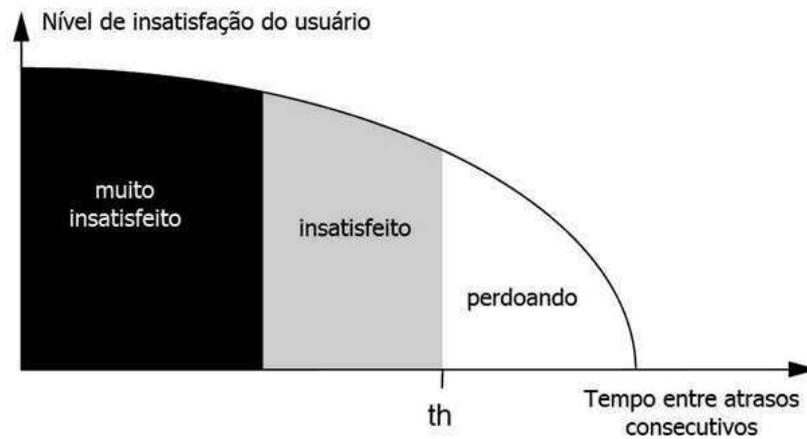


Figura 2.7: Ilustração da insatisfação do usuário.

Em um determinado problema de gerenciamento de energia, é interessante portanto, buscar um tempo  $th$  entre os atrasos, tal que, o usuário possa perdoar a ocorrência de atrasos consecutivos.

O objetivo do gerenciamento dinâmico de energia é minimizar as penalidades de desempenho e o consumo de energia. Esse objetivo pode ser alcançado com um gerenciador de energia ideal, que contemple o conhecimento do presente, passado e futuro da carga de trabalho [5]. No entanto, em sistemas computacionais de propósito geral, o conhecimento futuro da carga de trabalho é desconhecido a priori. Então, é necessário estimar o comportamento futuro da carga de trabalho, para que se possa tomar decisões de gerenciamento.

# Capítulo 3

## Metodologia da Pesquisa

A premissa fundamental para se aplicar de forma eficiente uma técnica de gerenciamento dinâmico de energia é que o sistema e seus componentes apresentem períodos de inatividade durante o tempo de execução. Também é importante que se possa prever, com um certo grau de confiança, as variações na carga de trabalho [20].

Dependendo do sistema computacional e de sua carga de trabalho, políticas de gerenciamento de energia projetadas utilizando técnicas distintas possivelmente tem desempenho diferentes [27]. Para interfaces de rede, Luiz et al. [17] propuseram uma política dinâmica de *timeout* considerando o limiar da taxa total de transferência de dados (*download* e *upload*) e um *timeout*.

Partindo-se desse pressuposto foi realizada a aquisição de cargas de trabalho geradas em diferentes ambientes computacionais e por usuários distintos, formando dessa forma um banco de dados. Assim, foi comparado o desempenho de políticas simuladas para diferentes cargas de trabalho.

### 3.1 Cargas de Trabalho

A primeira amostra, nomeada como Carga de Trabalho Pessoal foi obtida pelo uso livre da interface de rede, de um único usuário, realizando atividades de navegação, *downloads*, *uploads*, com intervalos de pouca atividade e longos períodos de ociosidade da interface de rede. A representação dessa carga de trabalho ao longo do tempo é apresentada na Figura 3.1.

O intervalo de amostragem da carga de uso livre Pessoal foi de 5 segundos. Foi realizada a aquisição dessa carga de trabalho durante aproximadamente 5.000 minutos.

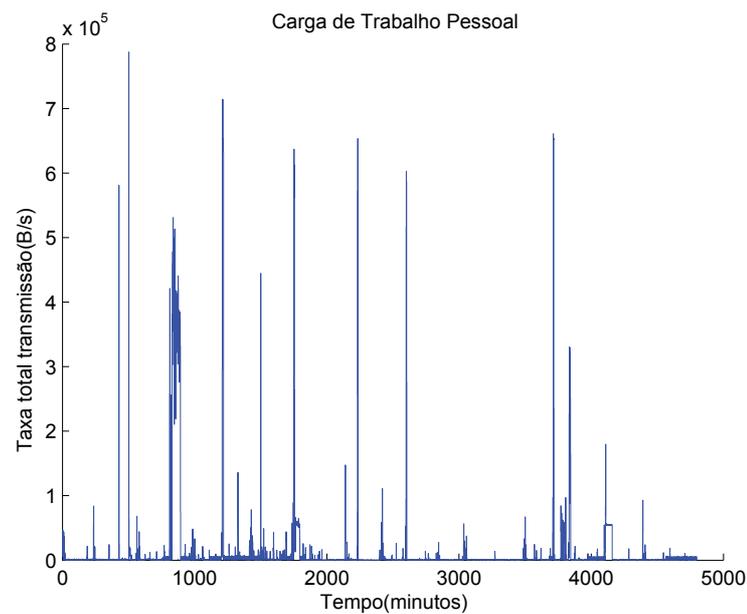


Figura 3.1: Transmissão total (*download* + *upload*) para uma carga de trabalho Pessoal.

A segunda amostra, nomeada como Carga de Trabalho Escritório, da mesma forma que a primeira, foi obtida pelo uso livre da interface de rede. No entanto, essa carga é composta pelo uso de vários usuários em diferentes ambientes computacionais, mas todos em ambiente de trabalho. A representação dessa carga de trabalho ao longo do tempo é apresentada na Figura 3.2.

O intervalo de amostragem da carga de trabalho de uso livre Escritório foi de 1 segundo. Foi realizada a aquisição dessa carga de trabalho durante aproximadamente 2.300 minutos.

A terceira amostra, nomeada como Carga de Trabalho Jogo, é uma carga gerada pela utilização de um jogo computacional online e pode ser caracterizada como uma carga de uso intensivo. Nessa carga de trabalho, há o uso da interface de rede pelo próprio jogo, assim como por um serviço de voz sobre IP usado para comunicação entre os jogadores. A representação dessa carga é apresentada na Figura 3.3.

O intervalo de amostragem da carga de trabalho de uso intensivo Jogo foi de 2 segundos. Foi realizada a aquisição dessa carga de trabalho durante aproximadamente 3.500 minutos.

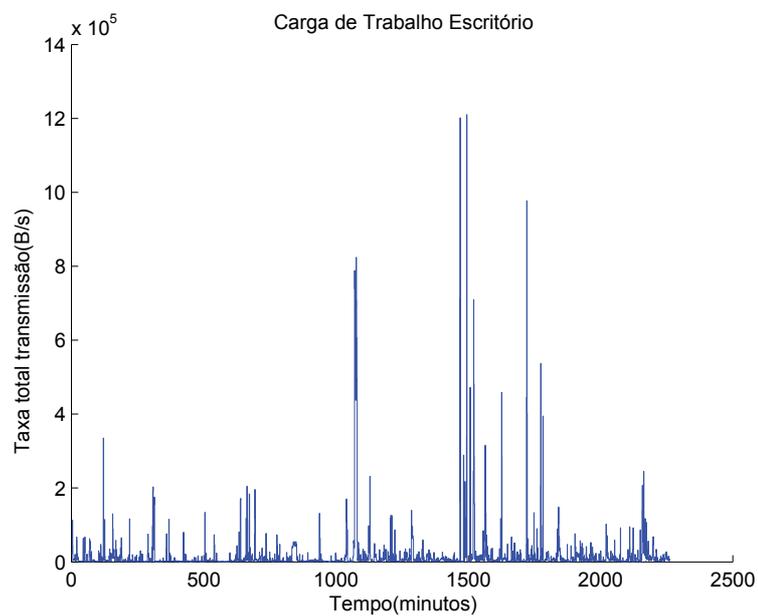


Figura 3.2: Transmissão total (*download* + *upload*) para uma carga de trabalho Escritório.

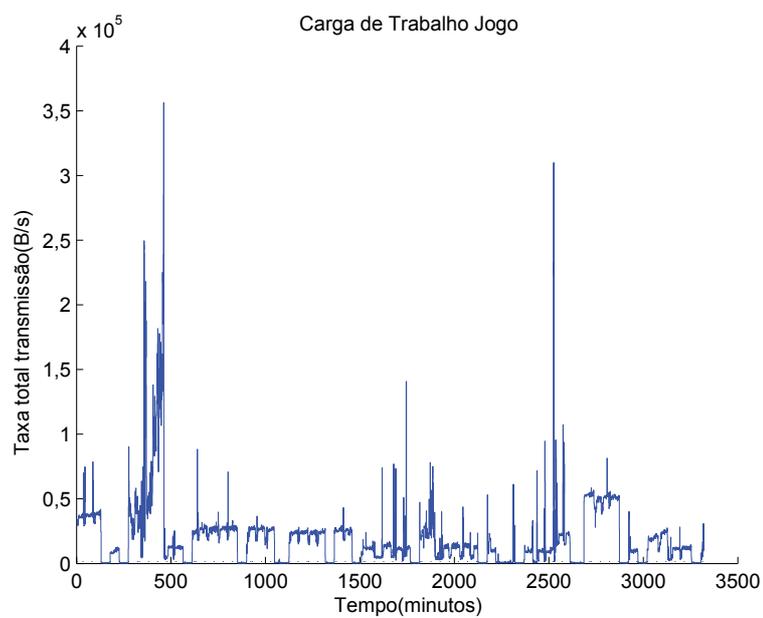


Figura 3.3: Transmissão total (*download* + *upload*) para uma carga de trabalho Jogo.

## 3.2 Definição do Limiar de Ociosidade da Interface de Rede

Como apresentado no Capítulo 2, obter o gráfico dos períodos ociosos versus períodos ativos pode ser fundamental para a implementação de determinadas políticas de gerenciamento de energia. Para melhor conhecimento das três cargas de trabalho apresentadas neste capítulo, foram traçados gráficos dos períodos ociosos versus períodos ativos afim de se verificar se estes possuem ou não a forma de um  $L$ .

Para tanto, é necessário que se defina um limiar na taxa total de transmissão de dados, para que se afirme com segurança que a interface está ou não ociosa. Isso porque há momentos em que o usuário não está utilizando a interface, mas alguns aplicativos, por exemplo: antivírus, podem estar executando alguma atualização em *background* fazendo com que a taxa de transmissão não seja nula. Nesse e em outros casos, o interesse é saber quando o usuário não está utilizando e só nesse caso permitir que a política de gerenciamento de energia atue. Na Figura 3.4 esse limiar é representado.

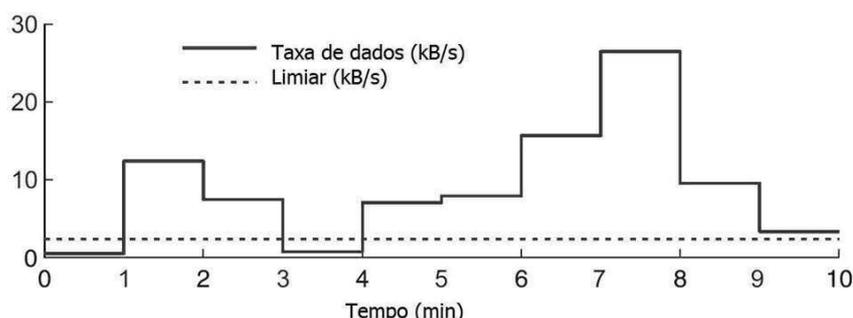


Figura 3.4: Limiar baseado na transmissão total (*download* + *upload*) para uma carga de trabalho.

O limiar de ociosidade da interface de rede foi definido a partir de experimentos em que o usuário informava se estava utilizando a interface de rede. Uma função custo, apresentada na Equação (3.1), foi implementada para definir o limiar que minimize o erro entre o estado estimado da interface de rede,  $a(t)$ , e o estado real informado pelo usuário,  $\hat{a}(t)$ . Para garantir que a interface está ociosa é necessário que o limiar seja igual ao menor valor calculado com a função custo [17].

$$J_{t_0, t_f}(\tau) = \sum_{t_0=1}^{t_f} \varepsilon^2(t) = \sum_{t_0=1}^{t_f} (a(t) - \hat{a}(t|\tau))^2 \quad (3.1)$$

Para definir esse limiar de inatividade para os principais ambientes computacionais utilizados na geração das cargas de trabalho coletadas, foram realizados experimentos

com variações da carga de trabalho e foi feita a aquisição das taxas de transmissão total *download + upload*, de forma semelhante ao procedimento utilizado por Luiz et al. [17]:

- (0 min, 1 min): intervalo ocioso;
- (1 min, 2 min): abrindo um email;
- (2 min, 3 min): conversando através de um serviço de mensagens;
- (3 min, 4 min): intervalo ocioso com o *browser* fechado;
- (4 min, 5 min): enviando um email;
- (5 min, 6 min): fechando o browser do email;
- (6 min, 7 min): intervalo ocioso;
- (7 min, 8 min): carregando e assistindo um vídeo na internet;
- (8 min, 9 min): carregando e assistindo um vídeo, com o email aberto;
- (9 min, 11 min): intervalo ocioso com o browser fechado;
- (11 min, 12 min): abrindo o email;
- (12 min, 13 min): intervalo ocioso com email aberto;
- (13 min, 14 min): entrando em uma rede social;
- (14 min, 15 min): intervalo ocioso com o browser da rede social aberto;
- (15 min, 16 min): abrindo um site de notícias;
- (16 min, 18 min): carregando e assistindo um vídeo na Internet;
- (18 min, 19 min): intervalo ocioso;
- (19 min, 20 min): fechando as páginas no *browser*;
- (20 min, 24 min): navegando em uma rede social;
- (24 min, 25 min): intervalo ocioso;
- (25 min, 30 min): navegando em uma página de notícias;
- (30 min, 32 min): intervalo ocioso;
- (32 min, 37 min): página da rede social aberta porém com a janela minimizada;

- (37 min, 45 min): página de notícias aberta porém com a janela minimizada;

Seguindo o procedimento utilizado por Luiz et al. [17] escolheu-se o menor valor do limiar que minimize uma função custo. Esse valor pode ser observado na Figura 3.5.

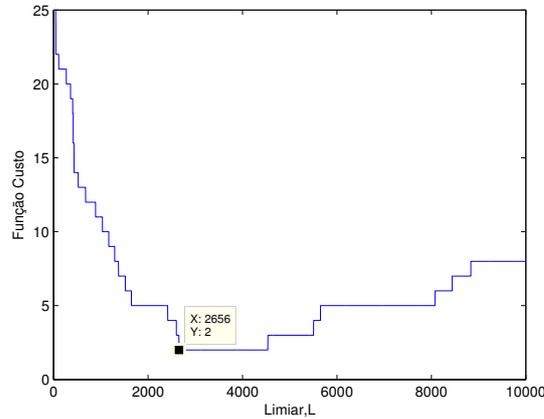


Figura 3.5: Função custo para a seleção do limiar de ociosidade da interface de rede.

Cada carga de trabalho foi obtida em computadores com especificações de *hardware*, sistemas operacionais e conjunto de *softwares* diferentes. Portanto, foram estimados limiares para cada sistema computacional. A partir disso, foi possível traçar o gráfico dos períodos ociosos versus períodos ativos. O resultado é apresentado na Figura 3.6.

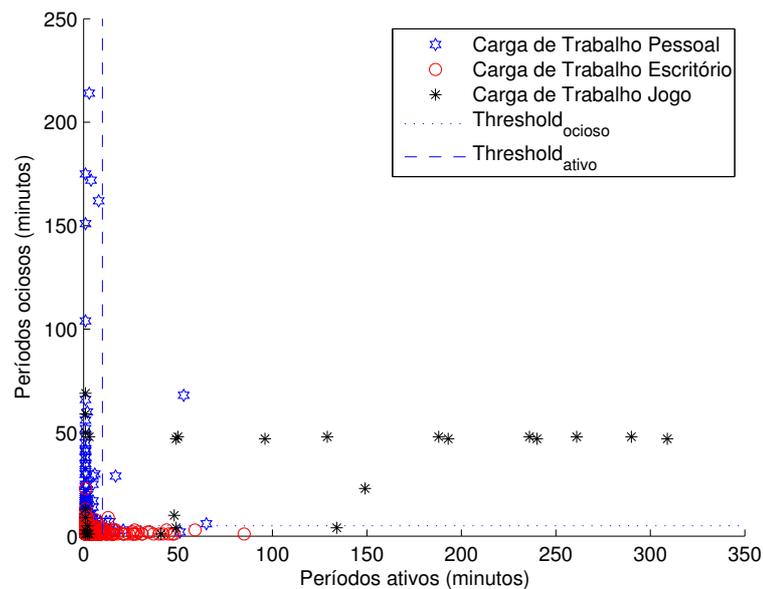


Figura 3.6: Curva dos períodos ociosos versus períodos ativos para as cargas de trabalho Pessoal, Escritório e Jogo.

A partir da observação da Figura 3.6, verifica-se que cada uma das três cargas de trabalho ocupa uma região distinta no gráfico. Assumindo que o limiar no eixo dos ‘períodos ociosos’ seja um  $Threshold_{ocioso}$ , e no eixo dos ‘períodos ativos’ seja um  $Threshold_{ativo}$ , é possível definir valores para o  $Threshold_{ocioso}$  e  $Threshold_{ativo}$  de forma que se possa identificar uma carga de trabalho como uma carga Pessoal, Escritório ou Jogo.

Para obter esse resultado variou-se o  $Threshold_{ocioso}$  de 1 a 350 e o  $Threshold_{ativo}$  de 1 a 350, englobando dessa forma do menor ao maior período ocioso e ativo, verificado em cada uma das três cargas de trabalho. Como é conhecido em qual área do gráfico cada tipo de carga de trabalho encontra-se majoritariamente, é possível definir esses limiares levando em consideração o maior número de acertos. Também é interessante que esses limiares sejam os menores possíveis. A curva que relaciona os valores de  $Threshold_{ocioso}$  e  $Threshold_{ativo}$  e a quantidade de acertos é apresentada na Figura 3.7.

Observando os resultados apresentado na Figura 3.7 é possível fazer algumas conclusões:

- A quantidade máxima de acertos possíveis seria de 639, pois existem 639 pares (intervalo ativos, intervalo ocioso) apresentados na Figura 3.6. As cargas de trabalho estão concentradas em diferentes regiões do gráfico, entretando, alguns raros pares (período ativo, período ocioso) aparecem em áreas onde a concentração majoritária é de uma outra carga de trabalho. Isso significa que a quantidade máxima de acertos possíveis deve ser inferior aos 639 pares.
- Escolhendo os valores do  $Threshold_{ativo} = 21$  e do  $Threshold_{ocioso} = 13$ , o percentual de acertos é maior que 97%.

### 3.3 Políticas Utilizadas

A partir do estudo realizado sobre políticas de gerenciamento de energia, quatro políticas foram simuladas em um *software* a fim de terem seu desempenho comparado para diferentes cargas de trabalho. Na Figura 3.8, é apresentado um diagrama de blocos que representa a simulação das políticas.

#### 3.3.1 Política de Timeout

Políticas de *timeout* são amplamente utilizadas, como já citado anteriormente nesse trabalho. No trabalho de Luiz et al. [17], uma política de *timeout* foi utilizada para o

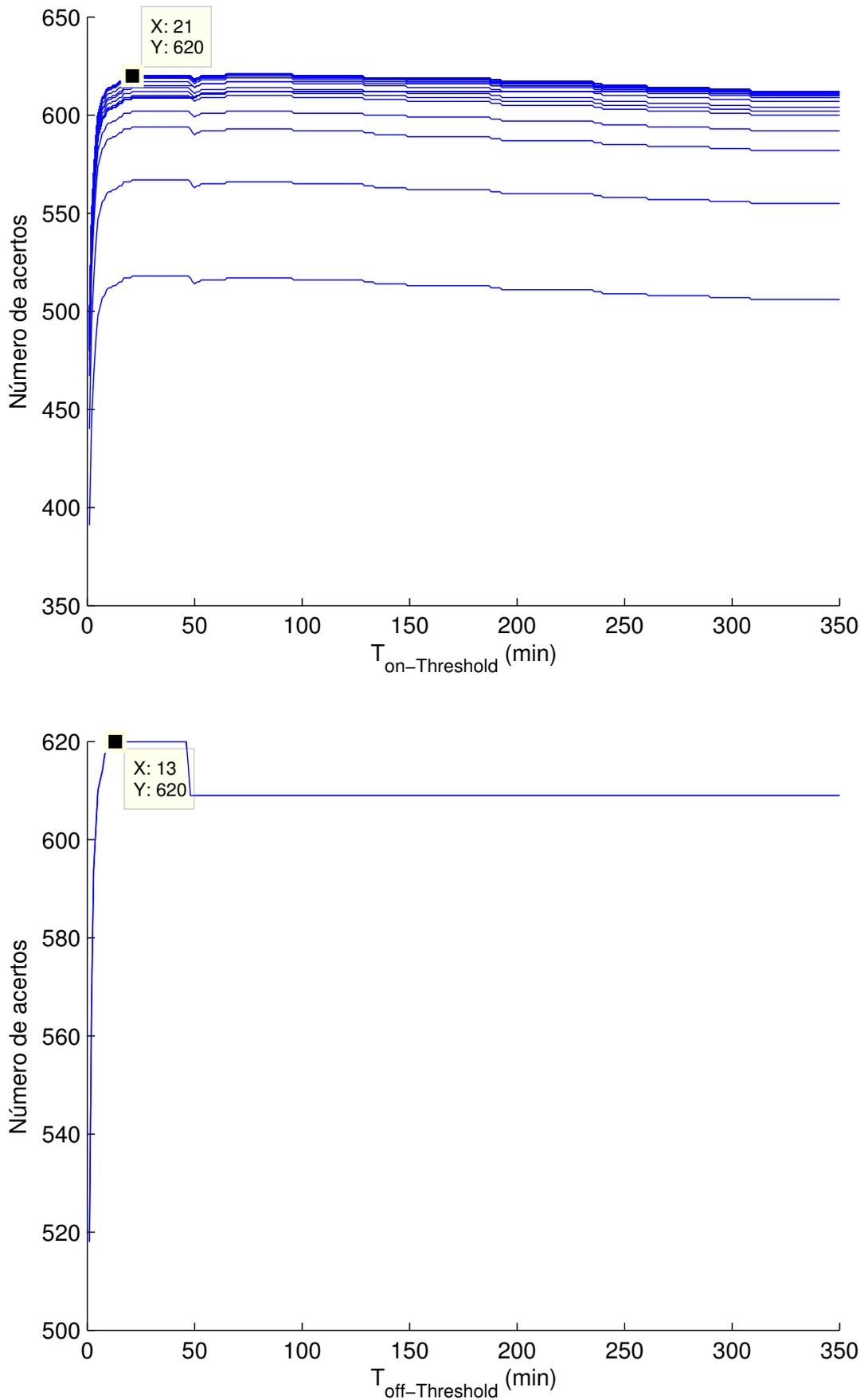


Figura 3.7: Relação entre os valores de  $Threshold_{ocioso}$  e  $Threshold_{ativo}$  e a quantidade de acertos.

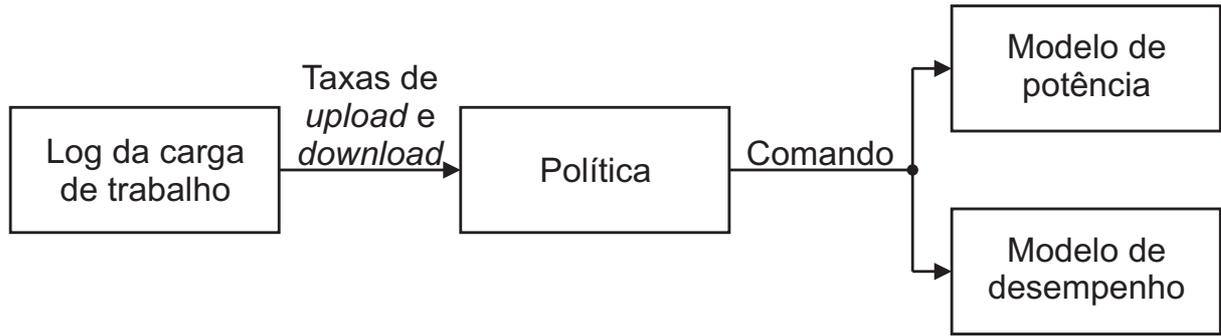


Figura 3.8: Diagrama de blocos para a simulação das políticas.

gerenciamento dinâmico de energia de uma interface de rede. A política foi analisada e validada por experimentos de medição de potência, apresentado bons resultados quanto a diminuição de potência consumida.

A política de *timeout* simulada nesse trabalho é semelhante à utilizada por Luiz et al. [17]. A política foi implementada em um *software* interativo. O pseudo-código que representa seu funcionamento é apresentado no Algoritmo 1. O valor do *timeout* é representado por  $\delta$ . Nesse algoritmo, a variável de entrada *taxa* corresponde a taxa total de transmissão de dados (*download* + *upload*). A saída, representada pela variável *comando*, refere-se ao comando que será enviado para desligar a interface, de acordo com o funcionamento da política.

---

**Algoritmo 1:** Algoritmo da Política de Timeout

---

**Dados:**  $\delta$ , *periodoOcioso* e *intervaloOcioso*

**Entrada:** *taxa*

**Saída:** *comando*

**se**  $taxa \leq limiar$  **então**

  └ *periodoOcioso* = 1 ;

**senão**

  └ *periodoOcioso* = 0 ;

**se** *periodoOcioso* == 1 **então**

  └ *intervaloOcioso* = *intervaloOcioso* + 1 ;

**senão**

  └ *intervaloOcioso* = 0 ;

**se** *intervaloOcioso* >  $\delta$  **então**

  └ *comando* = *desligar* ;

---

Essa política baseia-se na escolha de um valor ótimo do *timeout*,  $\delta$ , em minutos, que minimize o consumo de potência mantendo a penalidade de desempenho dentro de níveis aceitáveis. A interface de rede é desligada automaticamente quando ociosa por um período

maior que o valor do *timeout* estabelecido e o religamento só acontece quando ocorre uma requisição da carga de trabalho, em outras palavras, quando o usuário deseja utilizar a rede.

### 3.3.2 Política Preditiva

A política preditiva que foi simulada é baseada na apresentada por Hwang et al. [26]. O funcionamento desta política consiste em calcular uma estimativa para o próximo período ocioso, a partir dos períodos ociosos passados e das estimativas passadas. O desligamento da interface de rede ocorrerá apenas quando a valor estimado para o próximo período ocioso for maior que o valor do *break-even time*,  $T_{be}$ .

A política estima a duração do próximo período ocioso pela média acumulada dos períodos ociosos e estimados anteriores. A fórmula utilizada para calcular a estimativa é dada pela Equação (3.2). Como dito no Capítulo 2, o parâmetro  $a$  pode variar entre (0,1). A interface só será desligada se o intervalo estimado,  $I_{n+1}$ , for maior que o  $T_{be}$ . Dessa forma a política garante que a economia gerada pelo desligamento da interface compense o aumento do consumo gerado pela alteração de estados na interface de rede.

Diferente das demais políticas utilizadas nesse trabalho, a política Preditiva apresenta um religamento preditivo. Por isso, essa política deverá apresentar menores valores de penalidade de desempenho em comparação com as demais políticas. Isso porque, a interface deverá estar ativa sempre que ocorrer uma requisições da carga de trabalho. Para a implementação dessa política foi utilizada a fórmula de predição recursiva [26], representada na Equação (3.2).

$$I_{n+1} = a.i_n + (1 - a).I_n \quad (3.2)$$

Na simulação variou-se o parâmetro  $a$  de 0,1 até 1. O parâmetro  $a = 0$  não foi utilizado, pois a equação ficaria reduzida a  $I_{n+1} = I_n$ , ou seja, o valor estimado seria sempre o valor que a variável tivesse sido inicializada.

O pseudo-código que representa o funcionamento da política preditiva é apresentado no Algoritmo 2. Nesse algoritmo, a variável de entrada *taxa* corresponde a taxa total de transmissão de dados (*download* + *upload*). A saída, representanda pela variável *comando*, refere-se ao comando que é enviado para desligar ou religar a interface, de acordo com o funcionamento da política.

---

**Algoritmo 2:** Algoritmo da Política Preditiva

---

**Dados:**  $I_n, I_{n+1}, i_n, a, T_{be}, \text{periodoOcioso}$  e  $\text{intervaloOcioso}$ **Entrada:**  $\text{taxa}$ **Saída:**  $\text{comando}$ **se**  $\text{taxa} \leq \text{limiar}$  **então**└  $\text{periodoOcioso} = 1$  ;**senão**└  $\text{periodoOcioso} = 0$  ;**se**  $\text{periodoOcioso} == 1$  **então**└  $i_n = i_n + 1$  ;**senão**└  $i_n = 0$  ; $I_{n+1} = ai_n + (1 - a)I_n$  ;**se**  $I_{n+1} > T_{be}$  **então**└  $\text{comando} = \text{desligar}$  ;**se**  $I_{n+1} < \text{intervaloOcioso}$  **então**└  $\text{comando} = \text{ligar}$  ;

---

### 3.3.3 Política da Curva L

Como dito anteriormente, de acordo com Srivastava [14], existe uma relação entre os períodos ociosos e ativos. Lu e Micheli [6] usaram essa relação para definir uma política chamada de Política da Curva L. Ainda segundo Lu e Micheli essa política só se aplica à cargas de trabalho cujo a curva dos períodos ociosos versus períodos ativos possua o formato de  $L$ . Essa curva foi traçada para as três cargas de trabalho apresentadas nesse capítulo, e sua representação é apresentada na Figura 3.6. A política foi simulada em um *software* interativo. A principal diferença entre essa política e as demais, simuladas nesse capítulo, é que na política da curva L os desligamentos da interface ocorrem devido os intervalos ativos da carga de trabalho, e não pelos intervalos ociosos, que é característica das demais.

A duração do intervalo ativo é representado pela variável  $T_{thr}$ . No Algoritmo 3 é apresentado o pseudo-código que representa essa política. Nesse algoritmo, a variável de entrada  $\text{taxa}$  corresponde a taxa total de transmissão de dados ( $\text{download} + \text{upload}$ ). A saída, representanda pela variável  $\text{comando}$ , refere-se ao comando que é enviado para desligar a interface, de acordo com o funcionamento da política.

---

**Algoritmo 3:** Algoritmo da Política da Curva L

---

**Dados:** *periodoOcioso*, *intervaloOcioso*, *intervaloAtivo* e  $T_{thr}$ **Entrada:** *taxa***Saída:** *comando***se**  $taxa \leq limiar$  **então**└ *periodoOcioso* = 1 ;**senão**└ *periodoOcioso* = 0 ;**se**  $periodoOcioso == 0$  **então**└ *intervaloAtivo* = *intervaloAtivo* + 1 ;└ *intervaloOcioso* = 0 ;**senão**└ *intervaloOcioso* = *intervaloOcioso* + 1 ;└ *intervaloAtivo* = 0 ;**se**  $intervaloAtivo \geq T_{thr}$  **então**└ *comando* = *desligar* ;

---

### 3.3.4 Política Preditiva sem Religamento Automático

O religamento preditivo característico em uma política preditiva embora reduza a penalidade de desempenho pode comprometer o desempenho da política no critério ‘economia de energia’. Isso pode ocorrer em situações onde o religamento da interface seja demasiadamente antecipado. Nesse caso, o intervalo de tempo entre a interface ser religada e a chegada de uma requisição da carga de serviço, pode ser até maior que o intervalo em que a interface esteve desligada. Para verificar o desempenho da política preditiva sem o religamento automático, foi simulada em um *software* interativo, uma outra política preditiva com funcionamento semelhante a apresentada anteriormente, exceto no religamento automático. O pseudo-código que representa o funcionamento dessa política é apresentado no Algoritmo 4. Nesse algoritmo, a variável de entrada *taxa* corresponde a taxa total de transmissão de dados (*download* + *upload*). A saída, representada pela variável *comando*, refere-se ao comando que é enviado para desligar a interface, de acordo com o funcionamento da política.

---

**Algoritmo 4:** Algoritmo da Política Preditiva

---

**Dados:**  $I_n, I_{n+1}, i_n, a, T_{be}, \text{periodoOcioso}$  e  $\text{intervaloOcioso}$ **Entrada:**  $\text{taxa}$ **Saída:**  $\text{comando}$ **se**  $\text{taxa} \leq \text{limiar}$  **então**└  $\text{periodoOcioso} = 1$  ;**senão**└  $\text{periodoOcioso} = 0$  ;**se**  $\text{periodoOcioso} == 1$  **então**└  $i_n = i_n + 1$  ;**senão**└  $i_n = 0$  ; $I_{n+1} = ai_n + (1 - a)I_n$  ;**se**  $I_{n+1} > T_{be}$  **então**└  $\text{comando} = \text{desligar}$  ;

---

## 3.4 Simulações

A partir dos estudos sobre o gerenciamento dinâmico de energia, e conhecendo três cargas de trabalho distintas, foi possível implementar em um ambiente de simulação, políticas de gerenciamento de energia para interfaces de rede, e dessa forma realizar as primeiras comparações entre os desempenhos de cada política para cada carga de trabalho.

### 3.4.1 Simulação da Política de Timeout

Para a implementação de uma política de *timeout*, escolheu-se a política já implementada e validada por Luiz et al. [17]. A política foi otimizada de forma a contemplar o cálculo do  $T_{be}$ .

Na Figura 3.9 são apresentados os gráficos da potência média consumida versus o *timeout* para as três cargas de trabalho utilizadas. Para as três cargas de trabalho verifica-se que quanto maior o valor do *timeout* maior o valor da potência média consumida.

Na Figura 3.10 são apresentados os gráficos da Penalidade de desempenho versus o *Timeout* para as três cargas de trabalho utilizadas. As três cargas de trabalho apresentam um comportamento semelhante. Quanto maior o valor do *timeout*, menor a penalidade de desempenho. Isso ocorre pois um *timeout* grande implica em menor quantidade de desligamentos e consequentemente religamentos da interface de rede. Essa situação é

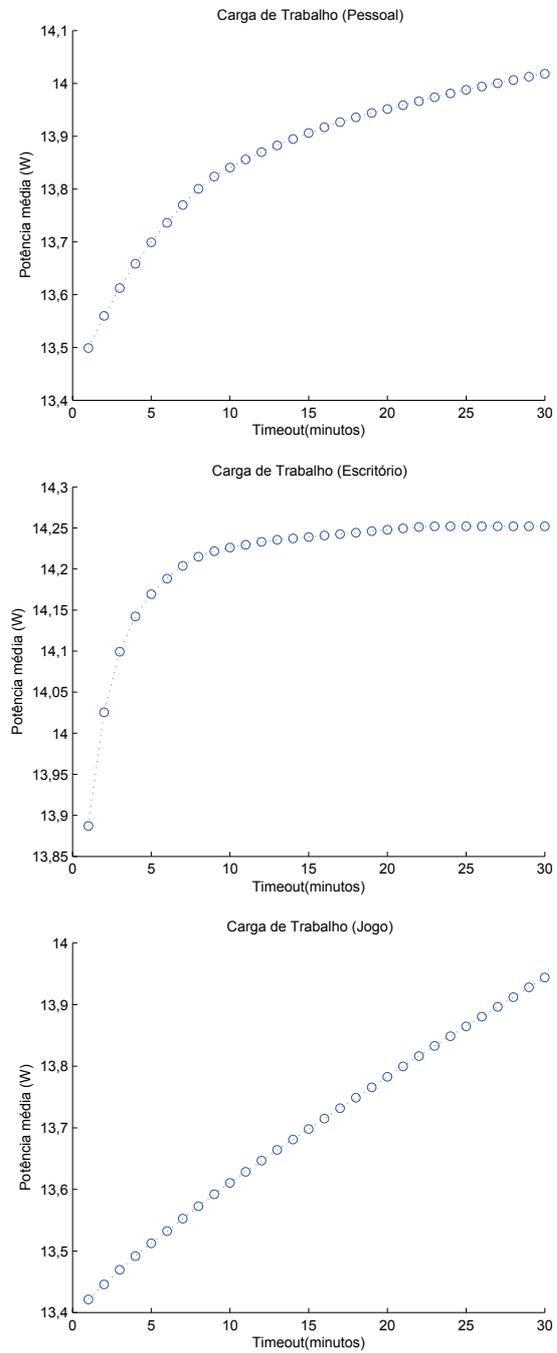


Figura 3.9: Curva da Potência média consumida versus *Timeout*, para as cargas de trabalho Pessoal, Escritório e Jogo.

visível na Carga de Trabalho Escritório na Figura 3.10. Como essa carga é caracterizada por curtos períodos de ociosidade, a escolha de um *timeout* maior de que 23 minutos já implica em nenhum desligamento. Conseqüentemente a penalidade de desempenho nesse caso é zero.

Na Figura 3.11 são apresentados os gráficos da Potência média versus a Penalidade de desempenho para as três cargas de trabalho utilizadas.

Para as três cargas, a penalidade de desempenho aumenta a medida que o valor de *timeout* diminui. Isso ocorre pois quanto menor é o valor do *timeout*, maior é o número de vezes em que a interface pode ser desligada. De forma geral, a carga de trabalho Escritório apresenta maiores valores de penalidade de desempenho quando comparada às demais. Isso ocorre pelo fato dessa carga de trabalho apresentar curtos períodos de ociosidade, essa característica pode ser observada na Figura 3.6 devido à localização dessa carga de trabalho. Com isso, a política atua com maior frequência e no entanto não economiza tanto quanto as demais.

### 3.4.2 Simulação da Política Preditiva

Para a simulação da política preditiva, o parâmetro  $a$  foi variado de 0.1 até 1. Quanto maior é o  $a$ , maior é o peso atribuído ao último período ocioso e conseqüentemente menor é o peso atribuído aos períodos ociosos estimados. O contrário também é verdadeiro.

Na Figura 3.12 são apresentados os gráficos da potência média versus o parâmetro  $a$ , para as três cargas de trabalho utilizadas.

Para a carga de trabalho Pessoal, verifica-se que, à medida que o parâmetro  $a$  cresce a potência média consumida também cresce. Isso mostra que para essa carga de trabalho os períodos ociosos passados são significativos para a previsão do período ocioso futuro. Já para a carga de trabalho Jogo, o contrário é verificado. Se desprezarmos os períodos ociosos passados e considerarmos apenas o último período ocioso, o desempenho dessa política é superior. A carga de trabalho escritório apresenta um comportamento um pouco diferente das demais. Nessa o melhor valor do parâmetro  $a$  é 0,6 o que de certa forma mostra que todos os períodos ociosos passados influenciam no período ocioso futuro. É importante ressaltar que a qualidade da estimação dos parâmetros não está sendo realizada com relação aos erros de estimação, mas sim em relação a uma medida indireta, que é a potência média.

Na Figura 3.13 são apresentados os gráficos da Penalidade de desempenho versus o Parâmetro  $a$ , para as três cargas de trabalho utilizadas.

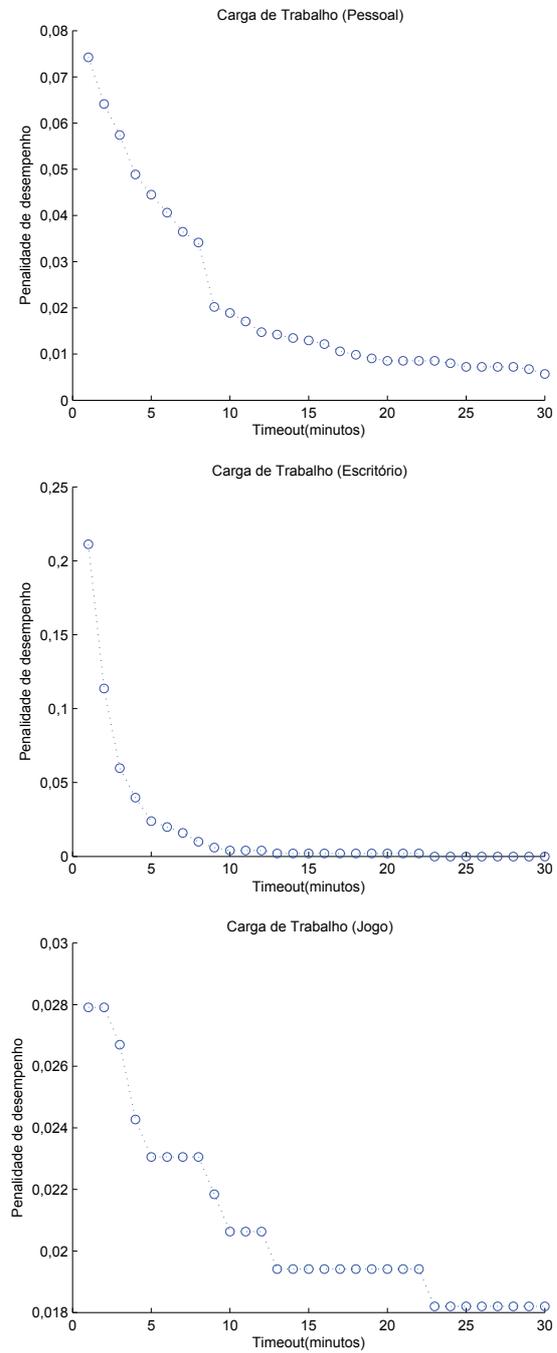


Figura 3.10: Política *Timeout*: curva da Penalidade de desempenho versus *Timeout*, para as cargas de trabalho Pessoal, Escritório e Jogo.

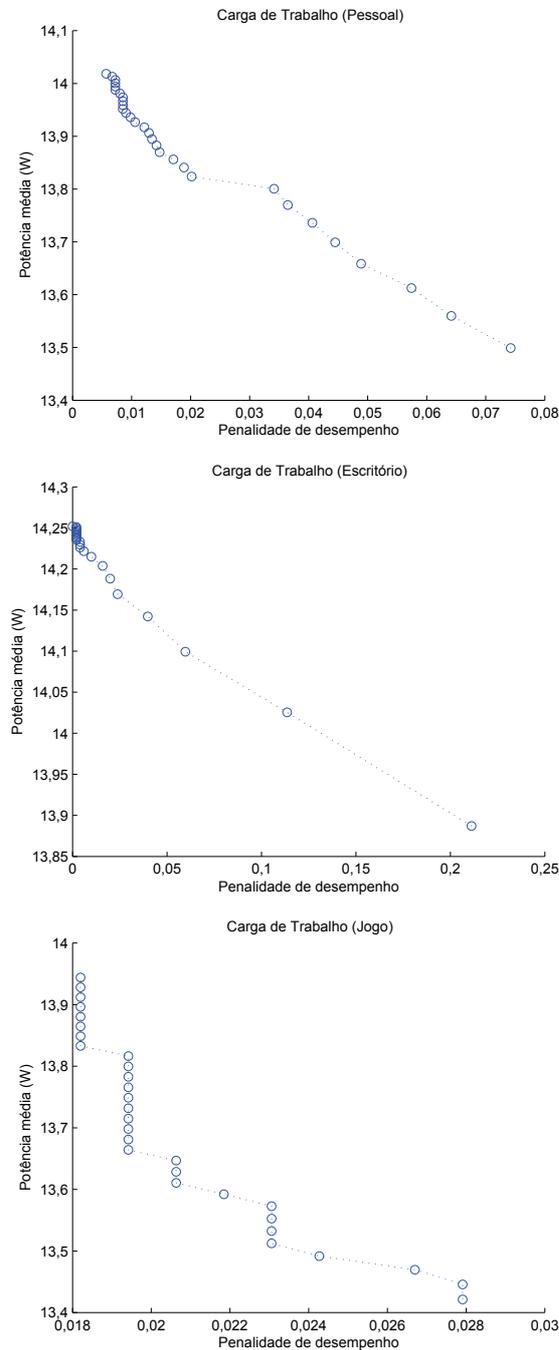


Figura 3.11: Política *Timeout*: curva da Potência média versus Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo.

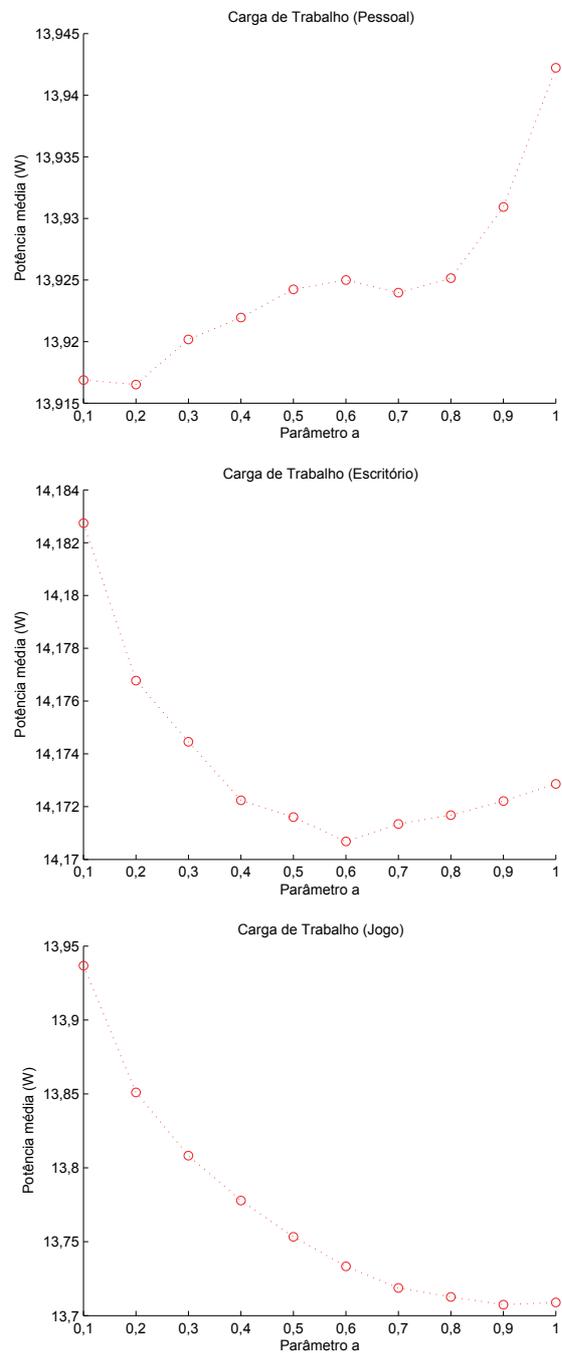


Figura 3.12: Política Preditiva: curva da potência média versus o parâmetro  $a$ , para as cargas de trabalho Pessoal, Escritório e Jogo.

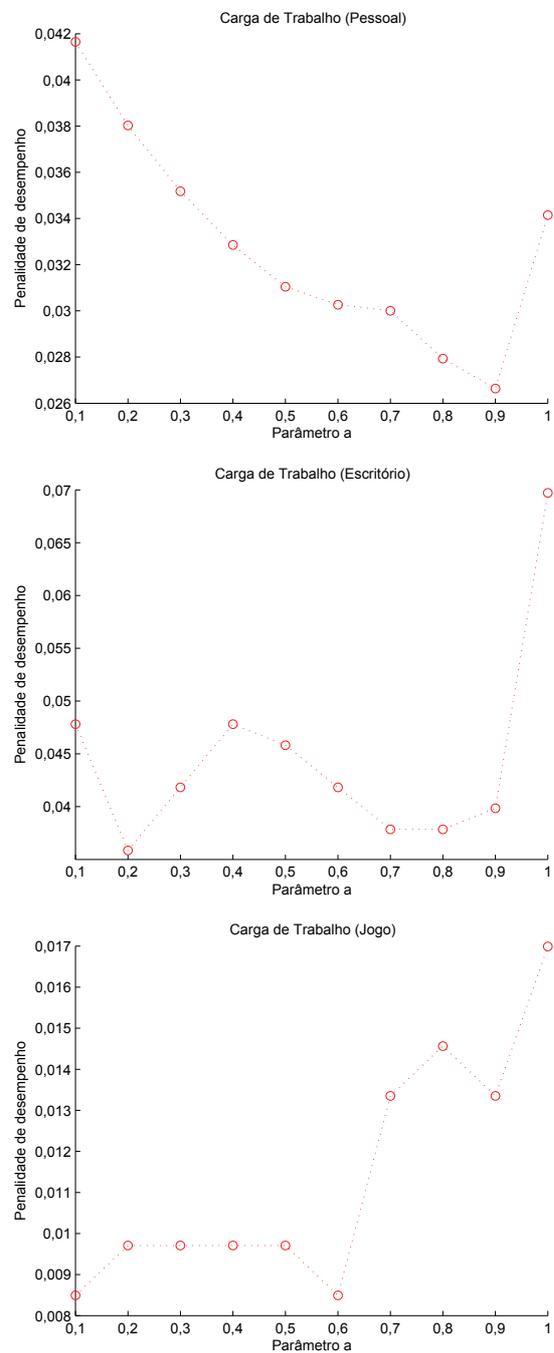


Figura 3.13: Política Preditiva: curva da Penalidade de desempenho versus o Parâmetro  $a$ , para as cargas de trabalho Pessoal, Escritório e Jogo.

Na Figura 3.14 são apresentados os gráficos da Potência média versus a Penalidade de desempenho para as três cargas de trabalho utilizadas.

Verifica-se que não existe uma associação diretamente proporcional entre a penalidade de desempenho e a potência consumida. Para uma mesma política, essa curva muda em função da carga de trabalho. Isso sugere a escolha de um parâmetro otimizado para cada carga de trabalho. Isso acontece porque a penalidade de desempenho na política preditiva ocorre quando o período ocioso estimado foi maior que o período ocioso real. Ou seja, quando ocorrer um desligamento, nem sempre ocorrerá também uma penalidade de desempenho.

### 3.4.3 Simulação da Política da Curva L

Para a simulação da política da Curva L, o *threshold* foi variado do menor ao maior valor do período ativo que ocorreu em cada uma das cargas de trabalho. Na Figura 3.15 são apresentados os gráficos da Potência média versus o *Threshold* para as três cargas de trabalho utilizadas.

Para as três cargas de trabalho, observa-se que a potência média diminui à medida que o *threshold* aumenta. Isso porque, quanto maior o *threshold*, maior a quantidade de intervalos ativos que são menores que o *threshold*, maior será a quantidade de intervalos ociosos que sucedem esses intervalos ativos e conseqüentemente maiores as chances da política atuar.

Na Figura 3.16 são apresentados os gráficos da Penalidade de desempenho versus o *Threshold* para as três cargas de trabalho utilizadas. Em todas as três cargas de trabalho observa-se um comportamento semelhante, a medida que o *threshold* cresce, a política atua mais vezes, desligando a interface, e a penalidade de desempenho aumenta.

Na Figura 3.17 são apresentados os gráficos da Potência média versus a Penalidade de desempenho para as três cargas de trabalho utilizadas. Quanto menor a potência consumida, maior é a penalidade de desempenho. Isso ocorre porque, à medida que o *threshold* aumenta, a política atua mais vezes. Isso faz com que a potência média diminua, mas também faz com que a penalidade cresça, pois a interface de rede é desligada e também religada mais vezes, acarretando em maiores penalidades de desempenho.

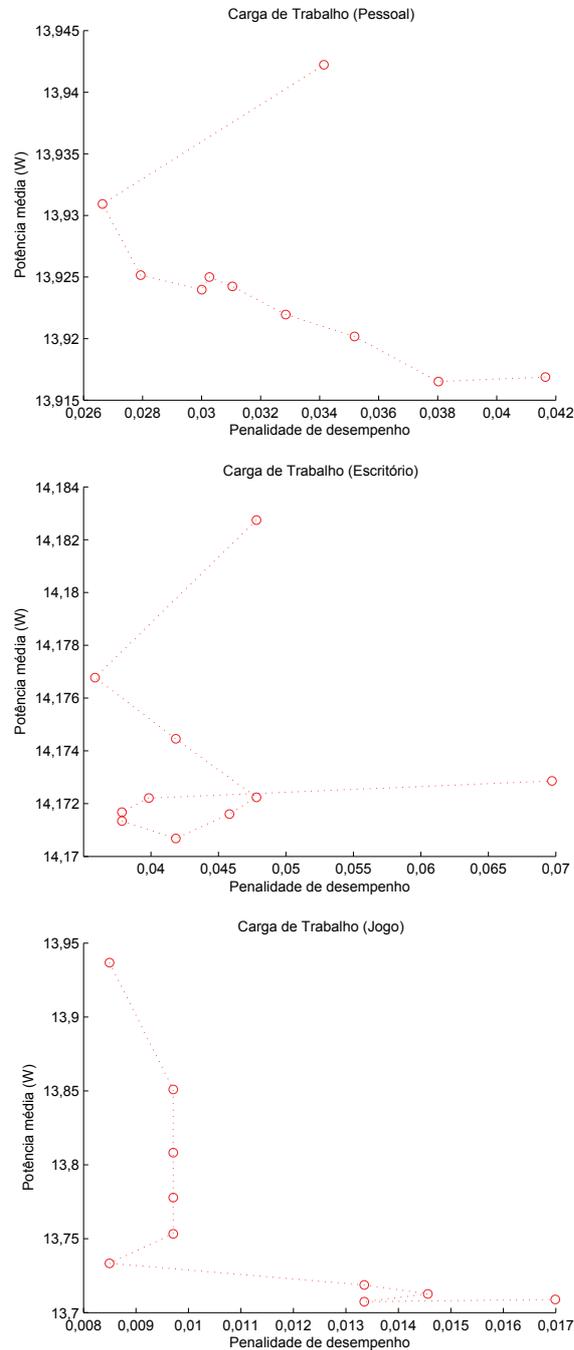


Figura 3.14: Política Preditiva: curva da Potência média versus a Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo.

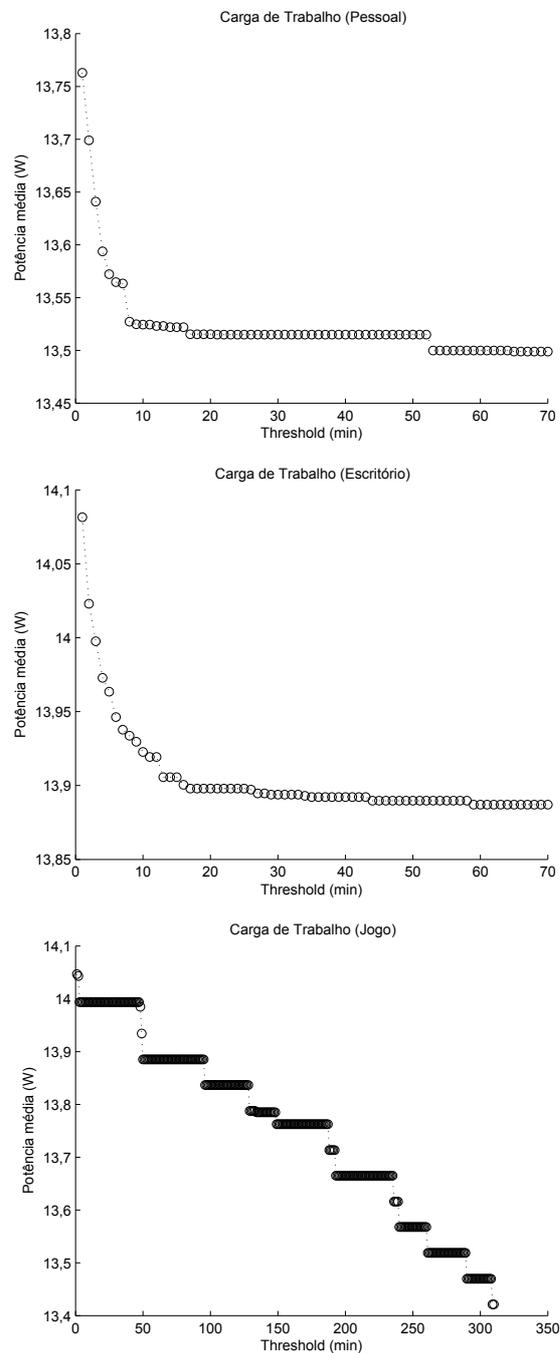


Figura 3.15: Política Curva L: curva da Potência média versus o *Threshold* para as cargas de trabalho Pessoal, Escritório e Jogo.

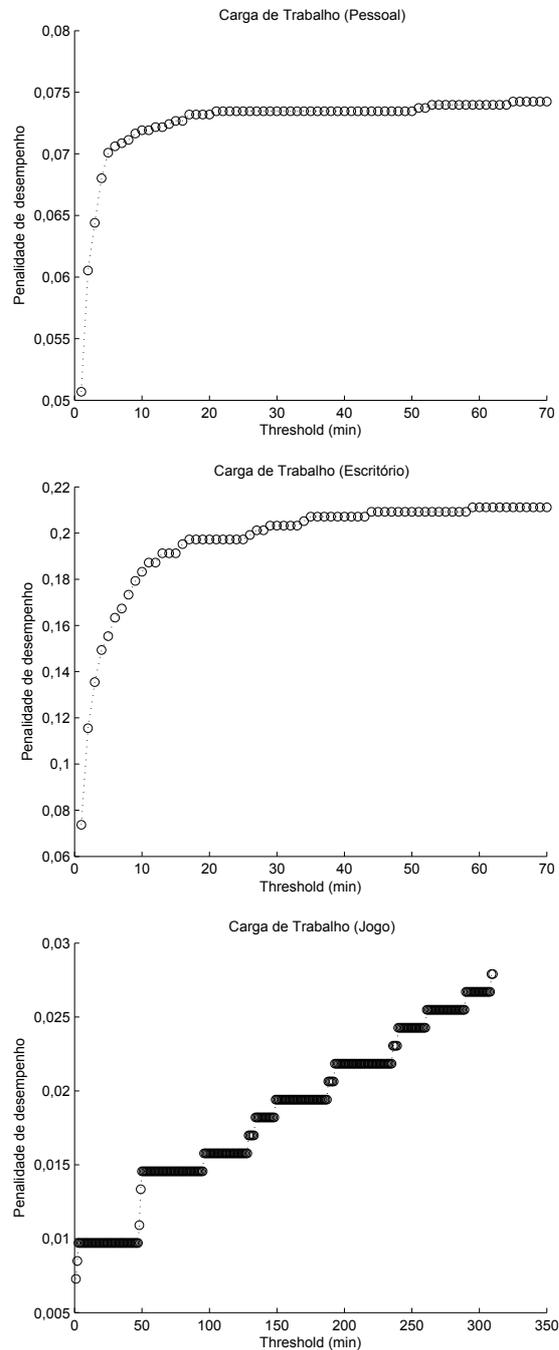


Figura 3.16: Política Curva L: curva da Penalidade de desempenho versus o *Threshold* para as cargas de trabalho Pessoal, Escritório e Jogo.

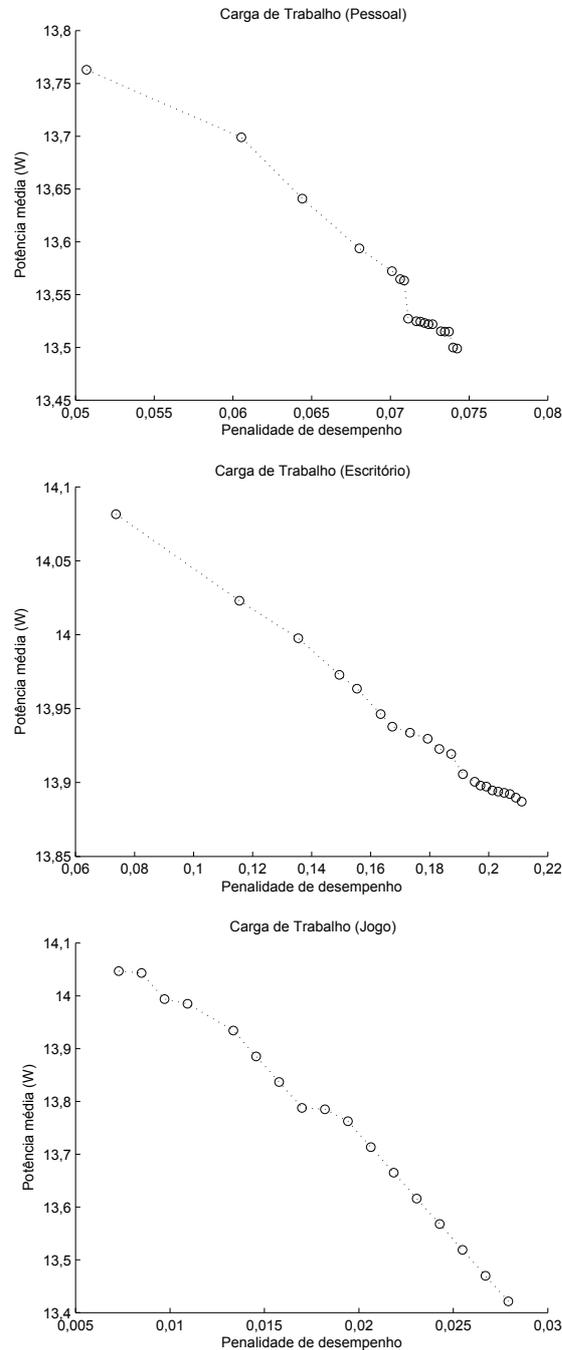


Figura 3.17: Política Curva L: curva da Potência média versus Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo.

### 3.4.4 Simulação da Política Preditiva sem Religamento Automático

Para a política preditiva sem religamento automático, os resultados das simulações podem ser verificados em seguida:

Na Figura 3.18 são apresentados os gráficos da Potência média versus o Parâmetro  $a$  para as três cargas de trabalho utilizadas. Para as três cargas de trabalho, pequenos valores do parâmetro  $a$  já possibilitam o menor valor de potência média consumida. Como o religamento da interface é manual, diferente da política preditiva, o parâmetro  $a$  influencia no 'quando desligar' mas não no tempo em que a interface ficará desligada. Isso comprova o que já era esperado: a política preditiva sem religamento automático alcança menores valores de potência média consumida quando comparada à política preditiva.

Na Figura 3.19 são apresentados os gráficos do Penalidade de desempenho versus o Parâmetro  $a$  para as três cargas de trabalho utilizadas.

A penalidade de desempenho aumenta com a quantidade de religamentos da interface que são necessários, para a carga de trabalho Pessoal e Escritório a partir do parâmetro  $a = 0,2$  e  $a = 0,3$ , respectivamente, o menor valor de potência média é alcançado. Isso significa que a política atuou todas as vezes que foi possível. A partir desse valor do parâmetro, a quantidade de desligamentos não aumenta, a potência média não diminui e conseqüentemente a penalidade de desempenho permanece constante. Para carga de trabalho Jogo, a política possui as mesmas oportunidades de atuar para todos os valores do parâmetro  $a$ . Isso ocorre devido as características dessa carga de trabalho. Assim sendo, para todos os valores do parâmetro  $a$  observa-se o mesmo valor de penalidade de desempenho.

Na Figura 3.20 são apresentados os gráficos da Potência média versus a Penalidade de desempenho para as três cargas de trabalho utilizadas.

Observa-se que a maior penalidade de desempenho acontece para a menor potência média consumida. Isso é o esperado pois a menor potência está associada à maior quantidade de desligamentos da interface e conseqüentemente a maiores valores de penalidade. A carga Jogo apresenta um comportamento um pouco diferente porque, para essa carga de trabalho, qualquer valor do parâmetro  $a$  já implica no menor valor de potência média, isso significa que, para todos os valores do parâmetro  $a$ , são realizadas as mesmas quantidades de desligamento da interface.

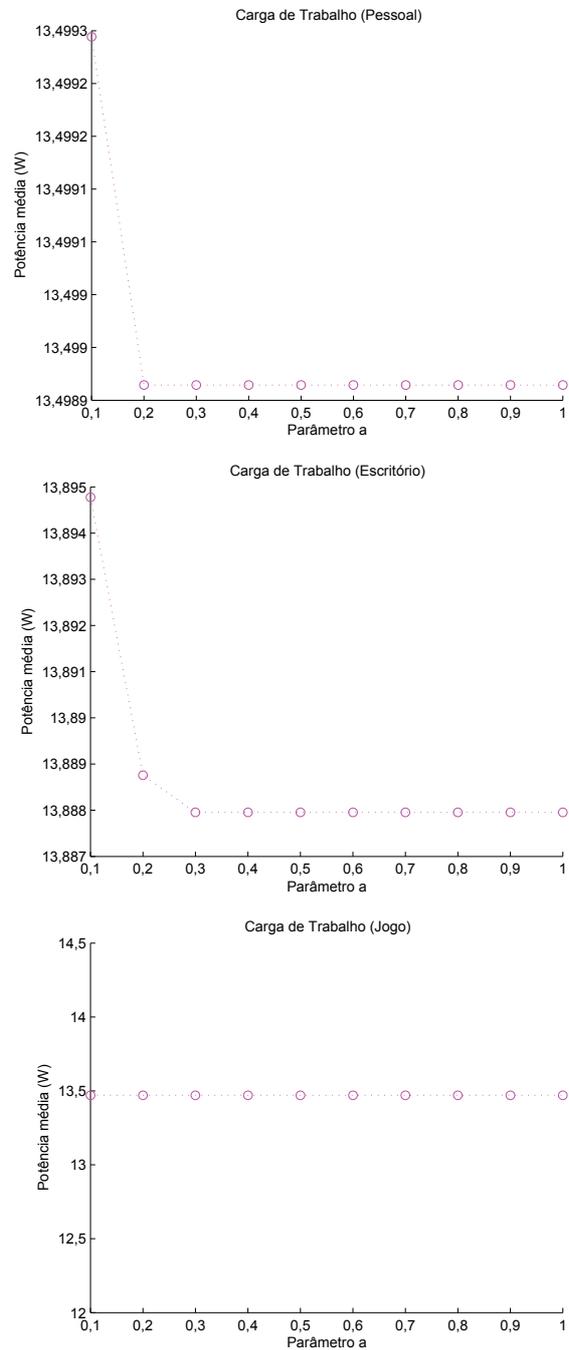


Figura 3.18: Política Preditiva SRA: curva da Potência média versus o Parâmetro  $a$  para as cargas de trabalho Pessoal, Escritório e Jogo.

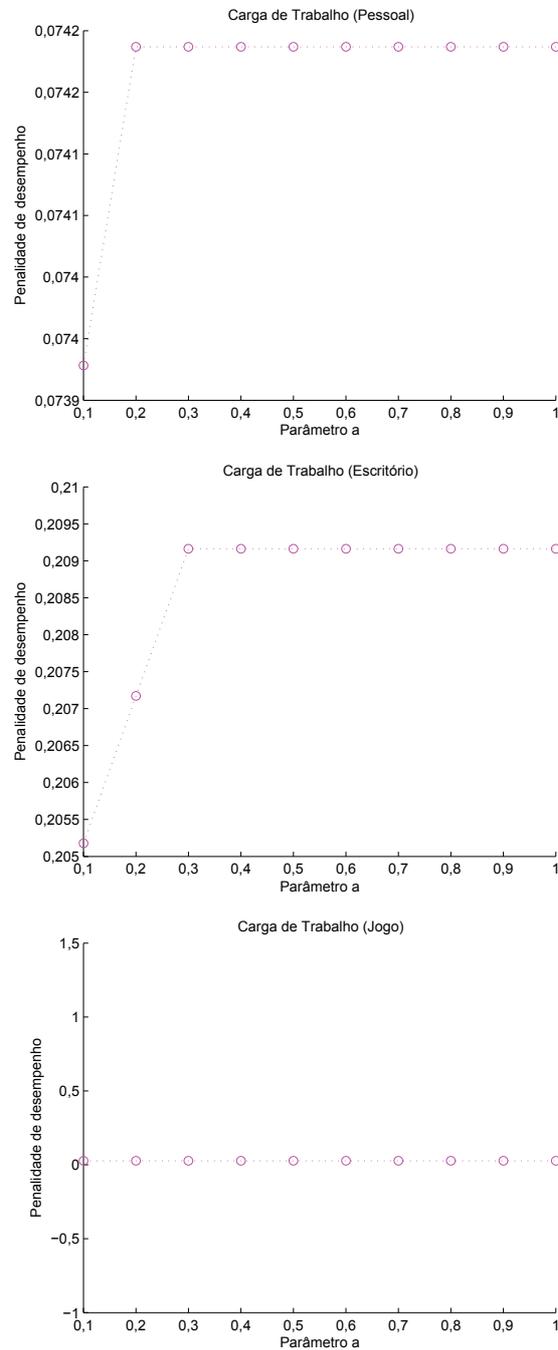


Figura 3.19: Política Preditiva SRA: curva da Penalidade de desempenho versus o Parâmetro  $a$  para as cargas de trabalho Pessoal, Escritório e Jogo.

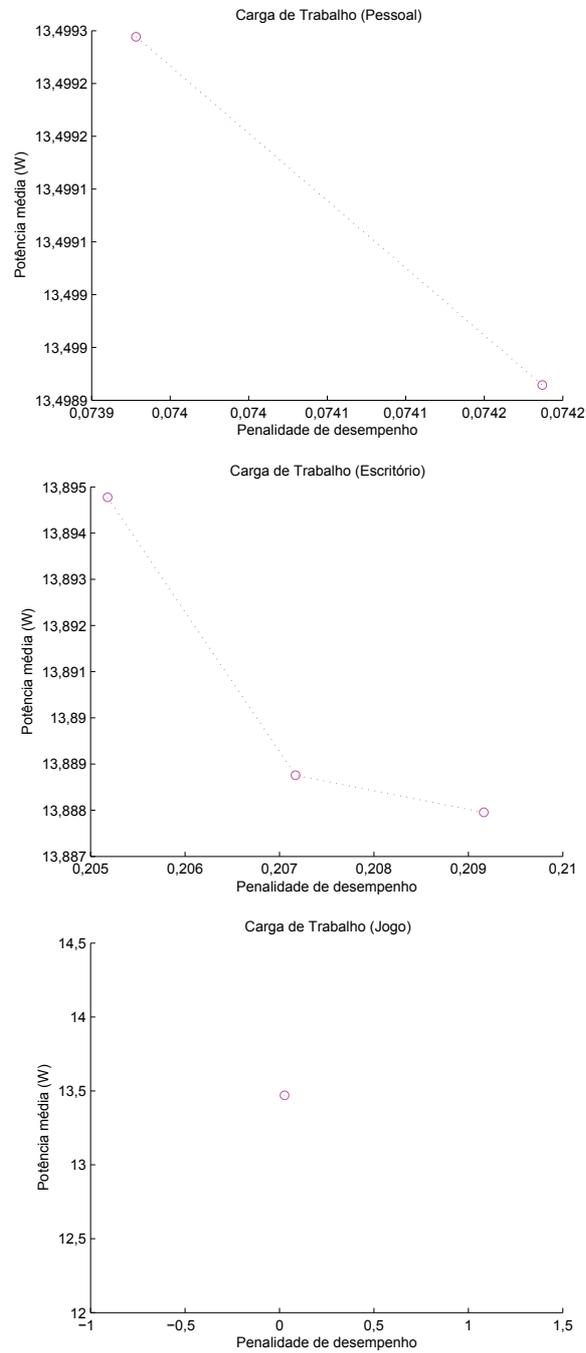


Figura 3.20: Política Preditiva SRA: curva da Potência média versus a Penalidade de desempenho para as cargas de trabalho Pessoal, Escritório e Jogo.

### 3.5 Curva de Pareto

Para comparar os desempenhos das quatro políticas para as três cargas de trabalho, foram traçados os gráficos da Potência média versus a Penalidade de desempenho, denominadas aqui de curvas de Pareto.

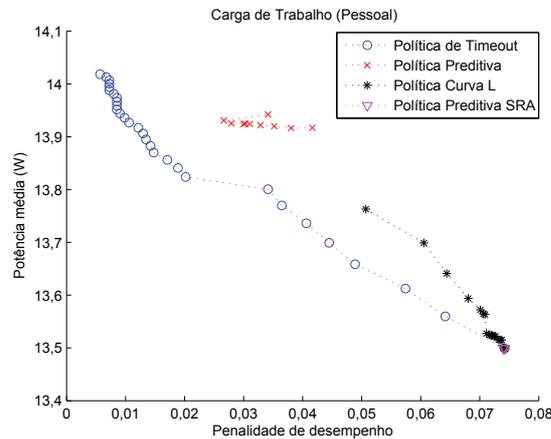


Figura 3.21: Curvas de Pareto para a carga de trabalho Pessoal.

A curva de Pareto para a carga Pessoal é apresentada na Figura 3.21. A política de *Timeout*, a política da Curva L e a política Preditiva SRA apresentam em seu melhor caso o mesmo valor de potência média consumida. Isso ocorre porque:

- A política de *timeout* alcança seu valor mínimo de potência para  $\delta = 1$ , ou seja, sempre que há um período ativo ocorre um desligamento
- A política da curva L alcança seu valor mínimo de potência para  $threshold = 70$ , ou seja, independente da duração do período ativo, a política irá desligar a interface quando ocorrer um período ocioso.
- A política preditiva SRA a partir do parâmetro  $a = 0,2$  já alcança o valor mínimo de potência, o que significa que, a partir desse valor do parâmetro, o comportamento dessa política foi semelhante às outras duas já faladas, desligando a interface sempre que inicia-se um período de ociosidade.

Já o comportamento da política preditiva difere-se das demais, pois o valor mínimo alcançado para a potência é superior ao das demais. Isso ocorre pela característica dessa política de ter um religamento preditivo. Muitas vezes a mesma religa a interface muito tempo antes do usuário querer usá-las. Isso tem a vantagem da penalidade de desempenho ser inferior a das demais, mas em contrapartida essa política não consegue economizar tanto quanto as demais.

A curva de Pareto para carga Escritório é apresentada na Figura 3.22. O comportamento das políticas para essa carga de trabalho é semelhante ao que acontece com a carga de trabalho Pessoal.

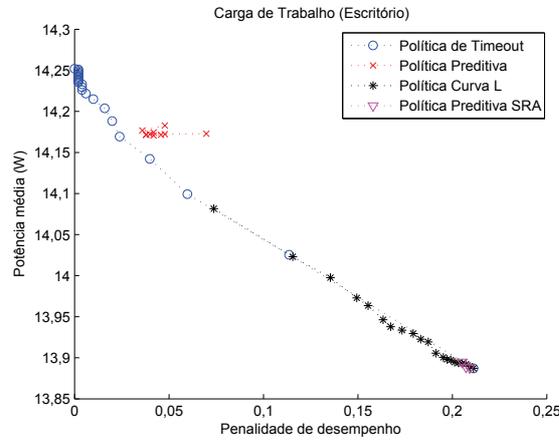


Figura 3.22: Curvas de Pareto para a carga de trabalho Escritório.

A curva de Pareto para carga Jogo é apresentada na Figura 3.23. Diferente das demais, para essa carga de trabalho, a política preditiva SRA não consegue atingir o mesmo valor de potência média consumida que a política de *timeout* e a política da Curva L alcançam.

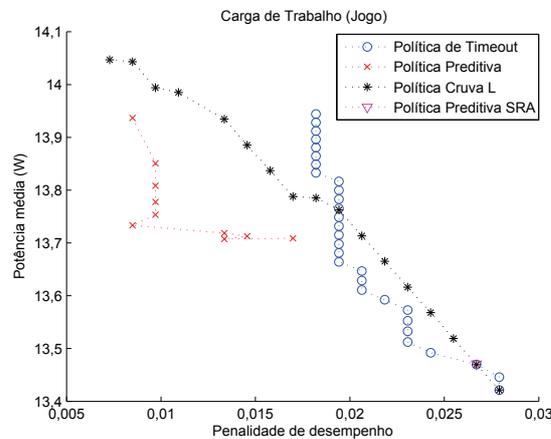


Figura 3.23: Curvas de Pareto para a carga de trabalho Jogo.

### 3.6 Simulação das Cargas de Trabalho Juntas

Foi verificado o desempenho das políticas para uma carga de trabalho Mista, ou seja, formada pela concatenação dos logs das cargas de trabalho Pessoal, Escritório e Jogo. A curva de Pareto para essa carga de trabalho pode ser visualizada na Figura 3.24.

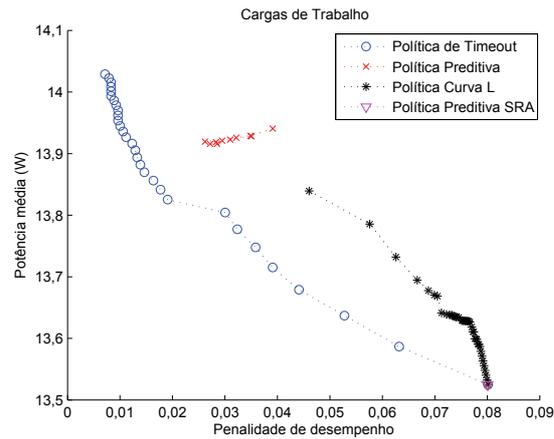


Figura 3.24: Curvas de Pareto para a carga de trabalho Mista.

Não foram verificadas diferenças significativas no desempenho de uma mesma política para essas três cargas de trabalho. Isso pode significar que:

- Essas cargas de trabalho embora apresentem comportamento diferente, ainda não representam uma variação significativa entre si.
- Os dados da plataforma experimental, em especial o baixo valor do  $T_{be}$ , favorecem que o desligamento da interface ocorra sempre que a mesma for dectada ociosa. Isso dificulta que as políticas apresentem um comportamento mais variado em relação a cada carga de trabalho.

O que é observado é que a mesma política pode apresentar melhor desempenho utilizando parâmetros diferentes para cargas de trabalho diferentes. Isso é verificado no desempenho da política Preditiva.

# Capítulo 4

## Resultados Experimentais

Nesse capítulo as políticas de gerenciamento dinâmico de energia que foram exaustivamente analisadas e otimizadas no Capítulo 3, terão agora seu desempenho comparado através da reprodução de um trecho da carga de trabalho Pessoal. Segundo Vilar et al. [16], durante o processo de desenvolvimento, uma política de energia deve ser testada várias vezes, até que se esteja certo de que a mesma está totalmente otimizada. Os testes devem ser realizados através da medição do consumo de potência de uma plataforma experimental submetida à cargas de trabalho específicas. Desta forma é possível comparar o desempenho de uma política de energia com outra.

A plataforma experimental utilizada nessa dissertação foi um *notebook* com Processador Intel Core i7 620M; máxima frequência de operação 2,66Hz; sistema Operacional Windows 7 Home Basic; memória RAM 8GB DDR3; Disco rígido de 640GB; monitor 14"(1366x768) TFT LCD; rede sem fio Azurewave AW-NE 104H e rede 10/100/1000 Mbps Gigabit Ethernet.

### 4.1 Medição da Tensão e Corrente Fornecidas ao *Notebook*

Para medir a tensão e corrente que são fornecidas ao *notebook*, foi utilizado um módulo de aquisição de dados NI USB-6210 conectado a um computador pessoal (PC) via *Universal Serial Bus* (USB), como ilustrado na Figura 4.1. No PC foi utilizado o programa LabVIEW no sistema operacional Windows.

Conforme a Figura 4.2, para medir o valor  $V$  da tensão entre os terminais positivo(+) e negativo(-) da fonte, o canal de entrada analógico A/2 do módulo de aquisição de

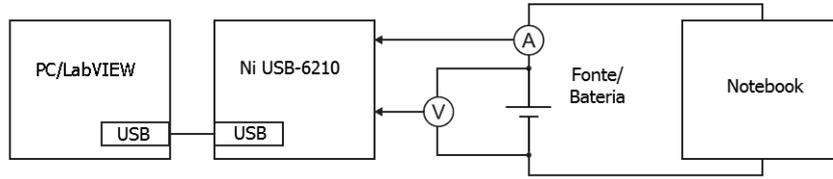


Figura 4.1: Diagrama de blocos para medição da tensão e corrente fornecidos ao *notebook*.

dados NI USB-6210 foi conectado ao terminal positivo da fonte, e o canal A/GND do NI USB-6210 foi conectado ao terminal negativo da fonte.

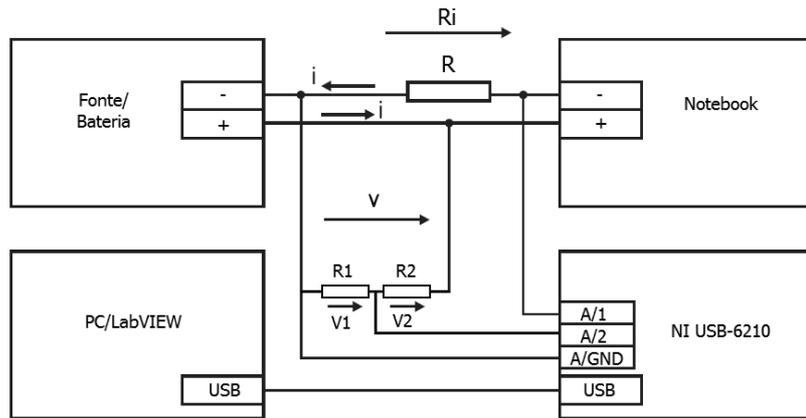


Figura 4.2: Diagrama elétrico para medição da tensão e corrente fornecidas ao *notebook* através do módulo de aquisição de dados NI USB-6210.

Para medir o valor  $i$  da corrente que flui do terminal negativo do notebook para o terminal negativo da fonte, foi colocada uma resistência  $R$  entre esses dois terminais. O canal A/1 no NI USB-6210 foi conectado ao terminal negativo do *notebook*. Assim o canal A/1 do NI USB-6210 mede a tensão  $Ri$  sobre a resistência  $R$ , que é proporcional à corrente  $i$ .

O diagrama de blocos para aquisição de dados no programa LabVIEW é apresentado na Figura 4.3. O bloco DAQ *Assistant* é o driver do LabVIEW com o NI USB-6210 e sua saída são os sinais de tensão e tensão no *shunt*. Esses sinais são exibidos no painel frontal do LabVIEW em valor instantâneo e como forma de onda, conforme a Figura 4.4. O bloco *Write Measurement File* salva em arquivo de texto esses valores.

Em um ambiente de simulação, o valor da resistência  $R$  é especificado, e os valores adquiridos no canal A/1 são automaticamente divididos por  $R$ , fornecendo assim os valores da corrente  $i$ . Isso é feito para que seja possível retirar a queda de tensão decorrente do uso da resistência  $R$ . Em seguida é calculada o valor da potência instantânea consumida durante a realização do experimento, como apresentado na Figura 4.5.



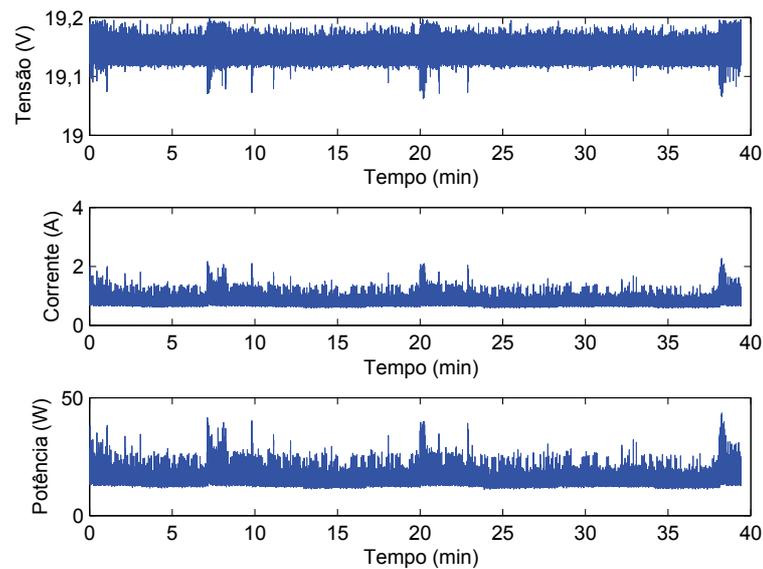


Figura 4.5: Valores de tensão, corrente e potência da plataforma experimental.

## 4.2 Estudo de Caso

Devido às limitações de tempo e maior viabilidade de reprodução, os experimentos para medição de potência foram realizados com a carga de trabalho Pessoal. O trecho apresentado na Figura 4.6 foi escolhido porque é representativo da carga de trabalho Pessoal, consistindo de curtos períodos de atividade seguidos de longos períodos de ociosidade. Para esse trecho, foram refeitas as simulações para cada política de trabalho e verificou-se que os valores de penalidade de desempenho eram aceitáveis. Isso pode ser verificado a partir da Figura 4.7.

Pretende-se dessa forma comparar os resultados de simulação com os experimentos de medição de potência.

### 4.2.1 Especificações dos Estados de Energia de uma Interface de Rede

Na Figura 4.8 estão representados as variações na potência média devido ao desligamento e o religamento da interface de rede.

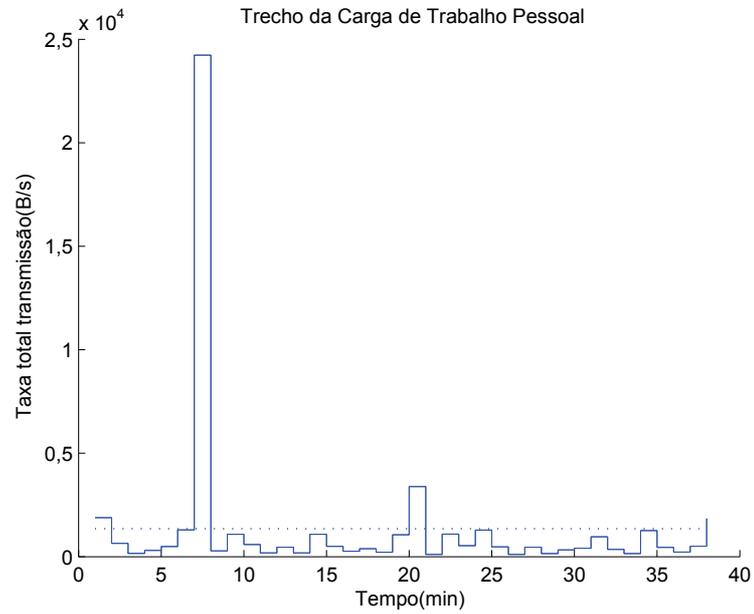


Figura 4.6: Taxa total de transmissão de dados para um trecho da carga de trabalho Pessoal.

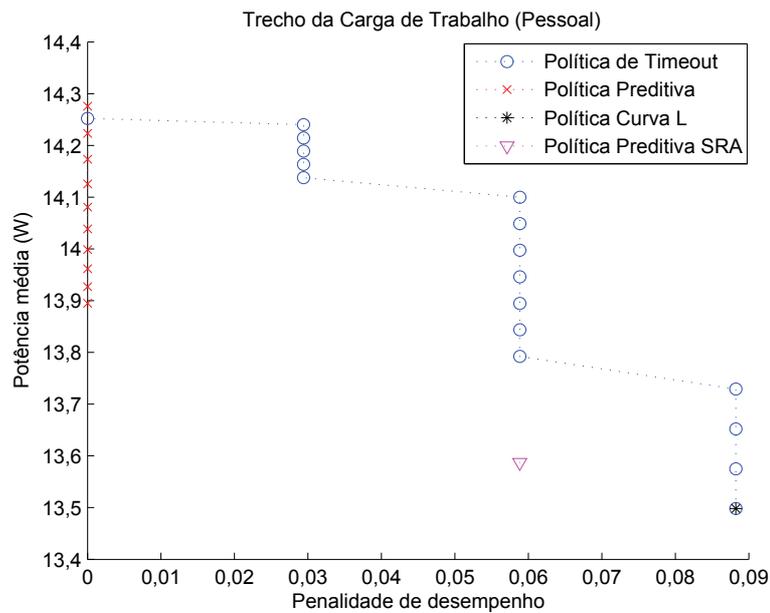


Figura 4.7: Simulação das Políticas para um trecho da Carga de Trabalho Pessoal

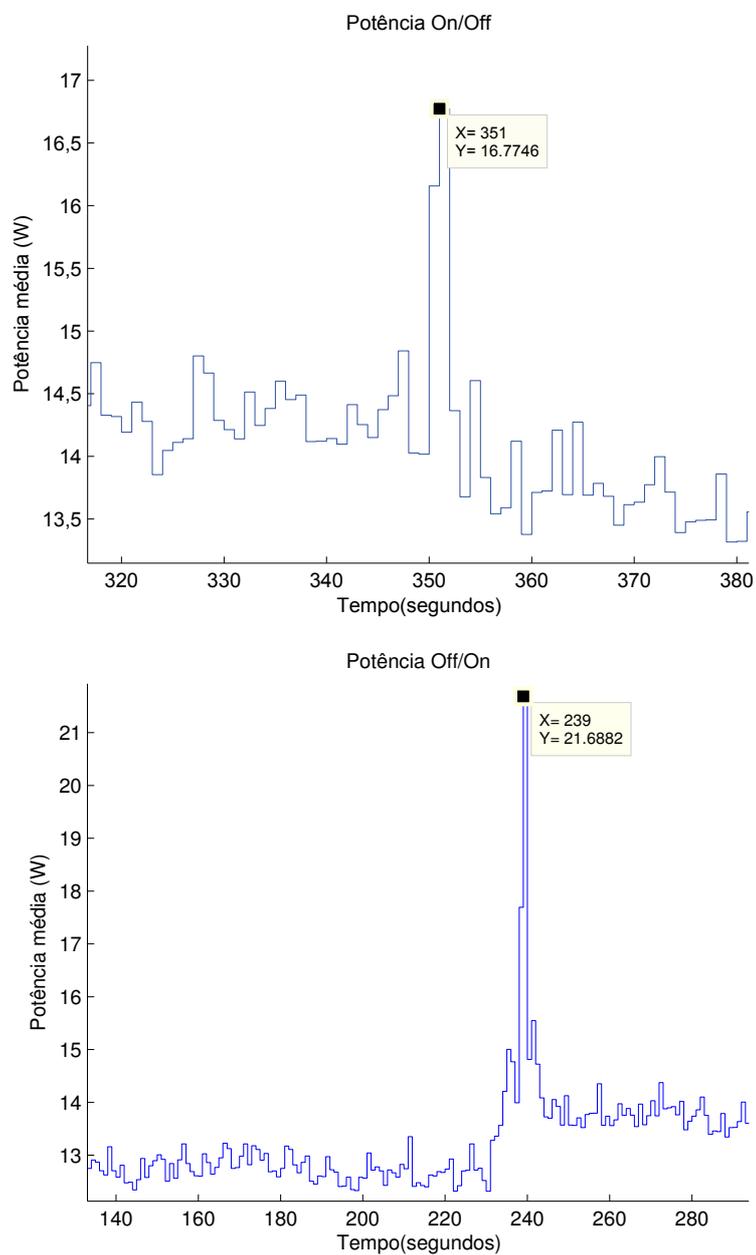


Figura 4.8: Transições dos estados da interface de rede ligada/desligada e desligada/ligada, respectivamente.

## 4.2.2 Roteiro dos Experimentos

O roteiro seguido para a realização dos experimentos de medição de potência da plataforma experimental foi o seguinte:

- Retirar a bateria do *notebook*;
- ligar o *notebook*;
- instalar o *software* com a política a ser testada;
- alterar os parâmetros das variáveis de acordo com os valores ótimos indicados pela simulação;
- desabilitar opções de esmaecer, suspender e desligar. Finalizar qualquer processo que não seja necessário ao funcionamento do *notebook*, e.g.: antivírus, skype...
- iniciar a medição de potência pelo *software* de aquisição de dados ao mesmo tempo em que iniciar a reprodução manual do trecho escolhido da carga de trabalho Pessoal.

É importante observar que, para as três políticas que não possuem o religamento preditivo da interface de rede, a mesma foi religada manualmente 15 segundos antes do instante em que seria necessária a utilização da interface. Isso foi feito para que o consumo gerado pela transição de estados estivesse dentro do intervalo contabilizado no cálculo da potência média consumida pela política instalada.

## 4.3 Resultados Experimentais

Foram realizados 5 experimentos, cada um de 39 minutos para cada uma das políticas estudadas, reproduzindo o trecho representado na Figura 4.6. Os intervalos ativos que ocorrem nos intervalos (1, 2), (7, 8), (20, 21) e (38, 39) minutos não foram contabilizados para o cálculo da potência média. Apenas os trechos em que a interface encontra-se ociosa, seja ligada ou desligada, é que serviram para o cálculo da potência média.

Os gráficos obtidos com a realização dos experimentos são apresentados na Figura 4.9. A linha tracejada em cada um dos gráficos representa a potência média consumida por aquela política de energia durante os intervalos em que a interface não estava sendo utilizada. Os valores medidos ficaram acima dos valores esperados. Mas isso é esperado dado que os valores utilizados para as simulação foram valores obtidos para o *notebook* sem nenhuma política de gerenciamento de energia instalada. Como a potência medida é

refente a todo o sistema e não só ao que a interface consome, verificamos um aumento no consumo devido ao processamento do *software* das políticas.

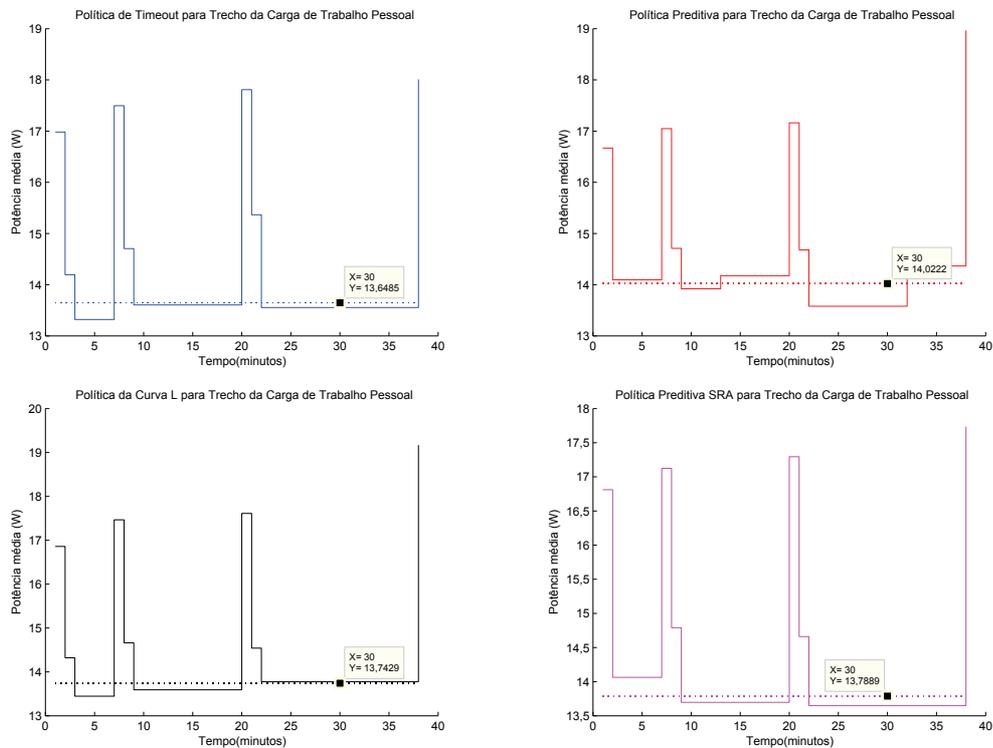


Figura 4.9: Resultados das medições de potência para cada uma das políticas testadas.

Na Tabela 4.1 são apresentados os valores médios das medições realizadas.

Tabela 4.1: Resultados dos experimentos.

Política	Penalidade de desempenho	Potência simulada	Potência medida
Timeout	0,08824	13,5W	13,6485W
Preditiva	0	13,89W	14,0222W
Curva L	0,08824	13,5W	13,7429W
Preditiva SRA	0,05882	14,25W	14,2966W
Sem Política	-	13,59W	13,7889W

# Capítulo 5

## Conclusões

Nesse trabalho foi proposta a comparação de várias técnicas de gerenciamento dinâmico de energia para interfaces de rede. Tomou-se como base para a implementação das políticas, os modelos de políticas de *timeout* e preditivas. Quatro políticas foram simuladas, otimizadas, e tiveram seu desempenho comparado através de medições de potência em uma plataforma experimental.

A partir de uma extensa coleta de dados foi possível indentificar três modelos de cargas de trabalho específicas para interfaces de rede. Foi aprimorada uma técnica que possibilitou distinguir uma carga de trabalho, a priori desconhecida, em uma das três cargas de trabalho apresentadas nessa dissertação.

Através das simulações foi verificado que nenhuma das políticas implementadass apresentou comportamento diferente quanto às cargas de trabalho. No entanto, uma mesma política pôde ser otimizada através da escolha dos seus parâmetros de acordo com a carga de trabalho a qual estava sendo submetida.

Os experimentos de medição de potência apresentaram os valores de consumo de potência já esperados para cada uma das políticas testadas.

### 5.1 Trabalhos Futuros

Dadas as limitações quanto à previsão do comportamento de uma carga de trabalho, ainda é preciso aperfeiçoar o mecanismo de estimação para as cargas de trabalho e relacionar os valores de penalidade de desempenho à satisfação do usuário.

Nas simulações de políticas de gerenciamento de energia para interfaces de rede, foram considerados valores de potência  $P_{on}$  e  $P_{off}$  relacionados a interface ligada e desligada.

Entretanto, a taxa total de transmissão (*download* + *upload*) influencia a potência. É preciso considerar a potência dinâmica gerada por essas taxas.

Pretende-se também investigar a alteração dos estados de potência do rádio, de forma a ser possível em algumas situações não apenas escolher entre ligado e desligado mas ter estados intermediários que possibilitem a economia de energia sem contudo realizar o desligamento da interface.

Após esses passos será possível propor técnicas de gerenciamento dinâmico de energia para dispositivos móveis que minimizem o consumo de potência maximizando a autonomia da bateria e a satisfação do usuário. Para isso, são propostos os seguintes trabalhos futuros:

1. Aperfeiçoar o mecanismo de estimação de atividade ou inatividade de uma interface de rede;
2. Implementar simulações considerando não somente as potências dos estados ligado e desligado mas também a potência dinâmica devido às taxas médias de *download* e *upload*;
3. Estender os modelos de cargas de trabalho;
4. Em sistemas interativos, correlacionar valores de penalidade de desempenho com os níveis de satisfação do usuário;
5. Propor técnicas de gerenciamento de energia para interfaces de rede em dispositivos que permitam não somente ligar e desligar a interface de rede, mas também alterar a potência do rádio;
6. Propor técnicas de gerenciamento de energia em dispositivos alimentados a bateria contemplando a minimização da potência e a maximização da autonomia da bateria.

# Referências Bibliográficas

- 1 KVETON, B.; GANDHI, P.; THEOCHAROUS, G.; MANNOR, S.; ROSARIO, B.; SHAH, N. Adaptive timeout policies for fast fine-grained power management. In: *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2*. AAAI Press, 2007. (IAAI'07), p. 1795–1800. ISBN 978-1-57735-323-2. Disponível em: <<http://dl.acm.org/citation.cfm?id=1620113.1620131>>.
- 2 LUIZ, S. O. D.; LIMA, A. M. N.; PERKUSICH, A. Modeling, estimation and control for power management in embedded systems. In: *Conf. Rec. DINCON'09*. Bauru, SP, Brasil: Anais do DINCON'09, 2009.
- 3 KIM, S.; KIM, H.; HWANG, J.; LEE, J.; SEO, E. An event-driven power management scheme for mobile consumer electronics. *IEEE Transactions on Consumer Electronics*, IEEE Computer Society, v. 59, p. 259–266, 2013.
- 4 RONG, P.; PEDRAM, M. Battery-aware power management based on markovian decision processes. In: *Conf. Rec. ICCAD '02*. New York, NY, USA: ACM Press, 2002. p. 707–713. ISBN 0-7803-7607-2.
- 5 CHUNG, E.-Y.; BENINI, L.; BOGLIOLO, A.; LU, Y.-H.; MICHELI, G. D. Dynamic power management for nonstationary service requests. *IEEE Trans. Comput.*, IEEE Computer Society, Washington, DC, USA, v. 51, n. 11, p. 1345–1361, 2002. ISSN 0018-9340.
- 6 LU, Y.-H.; MICHELI, G. D. Comparing system level power management policies. *IEEE Design & Test of Computers*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 18, n. 2, p. 10–19, 2001. ISSN 0740-7475.
- 7 LU, Y.-H.; BENINI, L.; MICHELI, G. D. Operating-system directive power reduction. In: *Conf. Rec. ISLPED '00*. New York, NY, USA: ACM, 2000. p. 37–42. ISBN 1-58113-190-9.

- 8 BENINI, L.; BOGLIOLO, A.; PALEOLOGO, G. A.; MICHELI, G. D. Policy optimization for dynamic power management. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, v. 18, n. 6, p. 813–833, Junho 1999. ISSN 0278-0070.
- 9 BROCK, B.; RAJAMANI, K. Dynamic power management for embedded systems. In: *Conf. Rec. of SOC Conference, 2003*. Portland, Oregon: Proceedings of the IEEE International SOC Conference, 2003. p. 416–419. ISBN 0-89791-964-5.
- 10 DHIMAN, G.; ROSING, T. Dynamic power management using machine learning. In: *Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on*. San Jose, CA: IEEE, 2006. p. 747–754. ISSN 1092-3152.
- 11 SIMUNIC, T.; BENINI, L.; GLYNN, P.; MICHELI, G. D. Event-driven power management. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 20, n. 7, p. 840–857, Jul 2001. ISSN 0278-0070.
- 12 VILAR, B.; SILVA, F.; LUIZ, S.; SILVA, J.; ALMEIDA, H.; PERKUSICH, A. Dynamic timeout power policy for network interfaces (no prelo). In: *Consumer Electronics - Berlin (ICCE-Berlin), 2014. ICCEBerlin 2014. IEEE Fourth International Conference on*. Piscatway: IEEE, 2014.
- 13 NEVES, B.; CRUZ, B.; MENDES, R.; BUBLITZ, F.; SILVA, J.; LUIZ, S.; PERKUSICH, A.; ALMEIDA, H. Computer systems power model estimation. In: *Consumer Electronics - Berlin (ICCE-Berlin), 2013. ICCEBerlin 2013. IEEE Third International Conference on*. Berlin: IEEE, 2013. p. 173–176.
- 14 SRIVASTAVA, M.; CHANDRAKASAN, A.; BRODERSEN, R. Predictive system shutdown and other architectural techniques for energy efficient programmable computation. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, v. 4, n. 1, p. 42–55, Março 1996. ISSN 1063-8210.
- 15 SUH, J.; KANG, D.-I.; CRAGO, S. Dynamic power management of multiprocessor systems. In: *Conf. Rec. WPDRTS '02*. Ft. Lauderdale, FL, USA: IEEE, 2002.
- 16 VILAR, B.; SILVA, A.; ULISSES, I.; SILVA, J.; PERKUSICH, A.; BUBLITZ, F. Estimating computer systems power consumption for time scaled experiments (no prelo). In: *Consumer Electronics - Berlin (ICCE-Berlin), 2014. ICCEBerlin 2014. IEEE Fourth International Conference on*. Piscatway: IEEE, 2014.
- 17 LUIZ, S.; PERKUSICH, A.; CRUZ, B.; NEVES, B.; ARAUJO, G. da S. Optimization of timeout-based power management policies for network interfaces.

- Consumer Electronics, IEEE Transactions on*, v. 59, n. 1, p. 101–106, Fevereiro 2013. ISSN 0098-3063.
- 18 WENG, L.-C.; WANG, X.; LIU, B. A survey of dynamic power optimization techniques. In: *System-on-Chip for Real-Time Applications, 2003. Proceedings. The 3rd IEEE International Workshop on*. NJ, USA: IEEE, 2003. p. 48–52.
- 19 HUANG, S.-M.; HU, C.; QI, L.-N. Predictive dynamic power management policy with consideration of outlying data. In: *Solid-State and Integrated Circuit Technology, 2006. ICSICT '06. 8th International Conference on*. Shanghai: IEEE, 2006. p. 2064–2066.
- 20 BENINI, L.; BOGLIOLO, A.; MICHELI, G. *A Survey of Design Techniques for System-Level Dynamic Power Management*. 2000. 299–316 p. Disponível em: <[citeseer.ist.psu.edu/benini00survey.html](http://citeseer.ist.psu.edu/benini00survey.html)>.
- 21 JIANG, Q.; XI, H.-S.; YIN, B.-Q. Adaptive optimisation of timeout policy for dynamic power management based on semi-markov control processes. *Control Theory Applications, IET*, v. 4, n. 10, p. 1945–1958, Outubro 2010. ISSN 1751-8644.
- 22 GREENAWALT, P. Modeling power management for hard disks. In: *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 1994., MASCOTS '94., Proceedings of the Second International Workshop on*. Durham, NC: IEEE, 1994. p. 62–66.
- 23 MADANI, N.; MASOUMI, M. A new optimization method for ctmdp system-level power management techniques. In: *Microelectronics, 2004. ICM 2004 Proceedings. The 16th International Conference on*. Piscataway: IEEE, 2004. p. 215–218.
- 24 LU, Y.-H.; MICHELI, G. D. Adaptive hard disk power management on personal computers. In: *VLSI, 1999. Proceedings. Ninth Great Lakes Symposium on*. Ypsilanti, MI: IEEE, 1999. p. 50–53. ISSN 1066-1395.
- 25 CHUNG, E.-Y.; BENINI, L.; MICHELI, G. D. Dynamic power management using adaptive learning tree. In: *Conf. Rec. ICCAD'99*. Piscataway, NJ, USA: IEEE Press, 1999. p. 274–279. ISBN 0-7803-5832-5.
- 26 HWANG, C.-H.; WU, A. C.-H. A predictive system shutdown method for energy saving of event-driven computation. *ACM Trans. Des. Autom. Electron. Syst.*, ACM, New York, NY, USA, v. 5, n. 2, p. 226–241, 2000. ISSN 1084-4309.

- 27 LUIZ, S. O. D. *Gerenciamento Dinâmico de Energia em Processadores com Cargas de Trabalho Variantes no Tempo*. Tese (Doutorado) — Universidade Federal de Campina Grande, Campina Grande, PB, BRASIL, 2012.