

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E  
DE COMPUTAÇÃO – PPGEEC**

**MODELAGEM APLICADA À ANÁLISE DE DESEMPENHO  
DO PROVEDOR DE SERVIÇOS DE UMA REDE DE  
AUTOMAÇÃO HOSPITALAR IEEE 802.3/PM-AH DE TEMPO  
REAL**

VINÍCIUS SAMUEL VALÉRIO DE SOUZA



NATAL – RN

2010

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

VINÍCIUS SAMUEL VALÉRIO DE SOUZA

**MODELAGEM APLICADA À ANÁLISE DE DESEMPENHO DO PROVEDOR DE  
SERVIÇOS DE UMA REDE DE AUTOMAÇÃO HOSPITALAR IEEE 802.3/PM-AH  
DE TEMPO REAL**

Proposta de dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

**Orientador:**

Prof. Dr. Gláucio Bezerra Brandão

**Co-orientador:**

Prof. Dr. Ricardo Alexsandro de Medeiros Valentin



NATAL

2010

Proposta de Dissertação de Mestrado **Aprovada** em \_\_\_\_ de \_\_\_\_\_ de 2010 pela banca examinadora composta pelos seguintes membros:

---

Prof. Dr. Gláucio Bezerra Brandão (Orientador)

---

Prof. Dr. Ricardo Aleksandro de Medeiros Valentim (Co-orientador)

---

Prof. Dr. Luiz Felipe de Queiroz Silveira (Avaliador Externo)

## **AGRADECIMENTOS**

Inicialmente, agradeço a Deus porque, como todo e bom cristão que tem fé, acredito que sem ele nada seria possível e, mesmo se fosse, nenhuma vitória teria graça alguma.

Agradeço a minha família, em especial meus pais, que sempre me deram apoio, acreditaram em mim e renunciaram muita coisa de suas vidas para que eu pudesse concluir a minha graduação e poder entrar no mestrado.

Agradeço a minha namorada Rochelly pelo amor, apoio, compreensão e carinho dado, os quais foram e serão extremamente importantes pra mim. Te amo Rochelly.

Agradeço ao professor Gláucio por ter acreditado na minha capacidade e ter aceitado orientar minhas atividades desde o tempo de graduação dando seguimento agora no mestrado.

Agradeço a professora Ana Maria por sempre estar presente nas minhas atividades acadêmicas e pelas suas contribuições que sempre levaram o meu trabalho pra frente.

Agradeço ao professor Luíz Felipe por ter aceitado o convite para participar da minha defesa de dissertação de mestrado e por sempre me ajudar quando precisei dele.

Agradeço ao professor Ricardo por estar sempre disponível quando precisamos dele, seja para debater sobre algum projeto, tirar dúvidas ou até mesmo fazer o café (isto é, quando alguém não tem se apoderado da cafeteira para fazer algum experimento).

Agradeço também à Cicília, Heliana, Léo, Heitor, Hélio, Diego, Vitor, Bruno, Marquinhos e todo o pessoal do LAHB (Laboratório de Automação Hospitalar e Bioengenharia) pelo apoio nas atividades que realizei desde o dia em que cheguei nesse laboratório.

## **RESUMO**

Com a incorporação de conceitos da automação industrial na área médica, surge a necessidade de estabelecer mecanismos que permitam a comunicação entre os dispositivos biomédicos de maneira segura e eficiente. Dentre as soluções para esse problema tem-se o PM-AH (Protocolo Multiciclos para Automação Hospitalar) que estabelece uma rede segmentada por leitos e coordenada por um elemento chamado Provedor de Serviços. O objetivo desse trabalho é modelar esse Provedor de Serviços e efetuar análises de desempenho sobre as tarefas realizadas pelo mesmo no estabelecimento e manutenção da rede hospitalar.

**Palavras-chave:** Dispositivos biomédicos, PM-AH, Provedor de Serviços, Modelagem e Análise de Desempenho.

## **ABSTRACT**

The incorporate of industrial automation in the medical are requires mechanisms to safety and efficient establishment of communication between biomedical devices. One solution to this problem is the MP-HA (Multicycles Protocol to Hospital Automation) that down a segmented network by beds coordinated by an element called Service Provider. The goal of this work is to model this Service Provider and to do performance analysis of the activities executed by in establishment and maintenance of hospital networks.

**Key-words:** Biomedical devices, MP-AH, Service Provider, Models and Performance Analysis.

## SUMÁRIO

1.	INTRODUÇÃO .....	17
1.1.	Objetivo Geral.....	20
1.2.	Objetivos Específicos .....	20
1.3.	Organização do Trabalho.....	20
2.	FUNDAMENTAÇÃO TEÓRICA .....	22
2.1.	PM-AH: Protocolo Multiciclos Para Automação Hospitalar.....	22
2.2.	Elementos do PM-AH .....	25
2.2.1.	Dispositivos biomédicos e leitos hospitalares .....	25
2.2.2.	Ciclos temporais: janelas síncronas e assíncronas.....	28
2.2.3.	Provedor de Serviços .....	30
2.3.	Mensagens do PM-AH .....	31
2.3.1.	<i>Request Message Join</i> (RMJ).....	31
2.3.2.	<i>Confirmation Message Join</i> (CMJ) .....	32
2.3.3.	<i>Message Token Pass</i> (MTP).....	33
2.3.4.	<i>Data Message</i> (DM) .....	34
2.3.5.	<i>Indication Beginning Window Assynchronous</i> (IBWA).....	34
2.3.6.	<i>Indication Alarm Message</i> (IAM) .....	35
2.3.7.	<i>Request Remove Join</i> (RRJ) .....	36
2.3.8.	<i>Request Control Message</i> (RCM) e <i>Confirmation Control Message</i> (CCM)..	36
2.3.9.	<i>Request Adjust Factor Message</i> (RAFM).....	37
2.3.10.	<i>Reindexing Message Node</i> (RMN) .....	38
2.3.11.	<i>Indication Synchronization Message</i> (ISM).....	39
3.	MODELAGEM DO PROVEDOR DE SERVIÇOS DE UMA REDE PM-AH.....	40
3.1.	Modelagem formal do banco de dados para o Provedor de Serviços .....	40
3.1.1.	Verificação formal da modelagem de persistência de dados do Provedor de Serviços.....	44
3.2.	Modelagem do Provedor de Serviços em Redes de Petri .....	51
3.2.1.	Modelo em Rede de Petri: mensagens de requisição de inscrição .....	53
3.2.2.	Modelo em Rede de Petri: mensagens trocadas em janelas síncronas .....	56
3.2.3.	Modelo em Rede de Petri: mensagens trocadas em janelas assíncronas .....	60
3.2.4.	Análise das propriedades da Rede de Petri do Provedor de Serviços.....	64
4.	ANÁLISE DE DESEMPENHO DO PROVEDOR DE SERVIÇOS.....	67
4.1.	Análise de desempenho do tempo de processamento do acesso aos registros: custo do processamento no banco de dados para efetuar o tratamento de requisições de inscrição.....	68
4.1.1.	Cenários de teste .....	69
4.1.2.	Resultados .....	69
4.2.	Análise de desempenho do Provedor de Serviços na rede PM-AH: transferência de mensagens de dados .....	78
4.2.1.	Cenários de teste .....	78
4.2.2.	Resultados .....	79
4.2.3.	Provedor de Serviços: Uma nova abordagem .....	88
5.	CONCLUSÃO .....	100
5.1.	Trabalhos em perspectiva .....	101
5.1.1.	Sistema supervisorio .....	101
5.1.2.	Implementação dos modelos em Rede de Petri.....	101



5.1.3. Reestruturação do PM-AH adicionando funcionalidades de Qualidade de Serviço (QoS).....	101
REFERÊNCIAS .....	103

## LISTA DE FIGURAS

Figura 1.1 - Exemplo de uma Rede PM-AH.....	19
Figura 2.1 - Esquema de filtragem de mensagens utilizando um switch com suporte a IGMP Snooping. (Fonte: Valentim, 2008) .....	23
Figura 2.2 - PM-AH na pilha de protocolos da Internet. (Fonte: Valentim, 2008) .....	24
Figura 2.3 - Elementos de uma rede PM-AH (Fonte: Valentim, 2008).....	25
Figura 2.4 – Esquema de associação de dispositivos em leitos. (Fonte: Valentim, 2008) .....	26
Figura 2.5 - Formato de uma RMJ.....	31
Figura 2.6 - Formato de uma CMJ.....	32
Figura 2.7 - Formato de uma MTP. ....	34
Figura 2.8 - Formato de uma DM.....	34
Figura 2.9 - Formato de uma IBWA.....	35
Figura 2.10 - Formato de uma IAM.....	35
Figura 2.11 - Formato de uma RRJ.....	36
Figura 2.12 - Formato de uma RCM/CCM.....	37
Figura 2.13 - Formato de uma RAFM.....	37
Figura 2.14 - Formato de uma RMN. ....	38
Figura 2.15 - Formato de uma ISM. ....	39
Figura 3.1 - DER do banco de dados para o Provedor de Serviços de uma rede PM-AH.....	42
Figura 3.2 - Modelo lógico para implementar o DER do banco de dados para o Provedor de Serviços da rede PM-AH usando um SGDB relacional. ....	43
Figura 3.3 – Tentativa de inserção de chaves duplicadas. (a) Tabela “Leitos” (b) Tabela “Dispositivos”.....	47
Figura 3.4 – Exemplo de preenchimento das tabelas “Dispositivos” e “Leitos” para realização de testes sobre o modelo de persistência de banco de dados. (a) Tabela “Dispositivos” (b) Tabela “Leitos”.....	48
Figura 3.5 – Resultados do teste de tentativa de violação de chave secundária. (a) Tentativa de violação da chave que referencia o dispositivo. (b) Tentativa de violação da chave que referencia o leito. ....	49
Figura 3.6 - Exemplo de preenchimento da tabela "Associações" para a realização dos testes. ....	50
Figura 3.7 - Resultados do teste de tentativa de remoção de registro nas tabelas cujas chaves primárias dos registros são referenciadas em outras tabelas. (a) Tentativa de remoção na tabela “Dispositivos” (b) Tentativa de remoção na tabela “Leitos”.....	50
Figura 3.8 - Rede de Petri do Provedor de Serviços de uma Rede PM-AH.....	52
Figura 3.9 - Rede de Petri referente ao processamento de mensagens de solicitação de inscrição de dispositivos. ....	54
Figura 3.10 - Rede de Petri referente ao processamento de mensagens trocadas durante uma janela síncrona. ....	57
Figura 3.11 - Rede de Petri referente ao processamento de mensagens trocadas durante uma janela assíncrona.....	61
Figura 3.12 - Análise das propriedades comportamentais do modelo completo em Rede de Petri do Provedor de Serviços. ....	65
Figura 4.1 – Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falharam em virtude dos dispositivos já estarem inscritos nos leitos.....	70

Figura 4.2 - Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falham em virtude dos leitos desejados estarem cheios. ....	72
Figura 4.3 - Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de mestre do leito. ....	73
Figura 4.4 - Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de escravo do leito. ....	75
Figura 4.5 - Tempos de processamento gasto na realização das operações de acesso ao banco de dados para todas as solicitações de inscrição realizadas. ....	77
Figura 4.6 - Atraso e <i>Jitter</i> da transmissão de dados (simulação para 10 leitos de 6 dispositivos cada). ....	81
Figura 4.7 - Atraso e <i>Jitter</i> da transmissão de dados (simulação para 25 leitos com 6 dispositivos cada). ....	83
Figura 4.8 - Atraso e <i>Jitter</i> da transmissão de dados (simulação para 50 leitos com 6 dispositivos cada). ....	85
Figura 4.9 - Atraso e <i>Jitter</i> da transmissão de dados (simulação para 60 leitos com 6 dispositivos cada). ....	87
Figura 4.10 - Comparativo do atraso e <i>jitter</i> na transmissão de dados entra a abordagem do PS sem passagem de <i>token</i> e com passagem de <i>token</i> (simulação para 10 leitos com 6 dispositivos cada). ....	91
Figura 4.11 - Comparativo do atraso e <i>jitter</i> na transmissão de dados entra a abordagem do PS sem passagem de <i>token</i> e com passagem de <i>token</i> (simulação para 25 leitos com 6 dispositivos cada). ....	93
Figura 4.12 - Comparativo do atraso e <i>jitter</i> na transmissão de dados entra a abordagem do PS sem passagem de <i>token</i> e com passagem de <i>token</i> (simulação para 50 leitos com 6 dispositivos cada). ....	95
Figura 4.13 - Comparativo do atraso e <i>jitter</i> na transmissão de dados entra a abordagem do PS sem passagem de <i>token</i> e com passagem de <i>token</i> (simulação para 60 leitos de 6 dispositivos cada). ....	97

## LISTA DE TABELAS

Tabela 2.1 - Descrição dos campos de uma RMJ.....	31
Tabela 2.2 - Descrição dos campos de uma CMJ.....	32
Tabela 2.3 - Descrição dos campos de uma MTP.....	34
Tabela 2.4 - Descrição dos campos de uma DM.....	34
Tabela 2.5 - Descrição dos campos de uma IBWA.....	35
Tabela 2.6 - Descrição dos campos de uma IAM.....	35
Tabela 2.7 - Descrição dos campos de uma RRJ.....	36
Tabela 2.8 - Descrição dos campos de uma RCM/CCM.....	37
Tabela 2.9 - Descrição dos campos de uma RAFM.....	37
Tabela 2.10 - Descrição dos campos de uma RMN.....	38
Tabela 2.11 - Descrição dos campos de uma ISM.....	39
Tabela 3.1 - Descrição dos atributos da entidade "Dispositivos".....	42
Tabela 3.2 - Descrição dos atributos da entidade "Leitos".....	42
Tabela 3.3 - Descrição dos atributos do relacionamento "Associações".....	42
Tabela 3.4 - Descrição dos atributos do relacionamento "Dados".....	42
Tabela 3.5 - Descrição dos atributos do relacionamento "Alarmes".....	43
Tabela 4.1 - Medidas de desempenho sobre os tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falharam em virtude dos dispositivos já estarem inscritos nos leitos.....	70
Tabela 4.2 - Medidas de desempenho sobre os tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falham em virtude dos leitos desejados estarem cheios.....	72
Tabela 4.3 - Medidas de desempenho referentes aos tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de mestre do leito.....	73
Tabela 4.4 - Medidas de desempenho referentes aos tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de mestre do leito.....	75
Tabela 4.5 - Medidas de desempenho dos tempos de processamento gasto na realização das operações de acesso ao banco de dados para todas as solicitações de inscrição realizadas.....	77
Tabela 4.6 - Parâmetros de desempenho da transmissão de dados (simulação para 10 leitos de 6 dispositivos cada).....	81
Tabela 4.7 - Parâmetros de desempenho da transmissão de dados (simulação para 25 leitos com 6 dispositivos cada).....	83
Tabela 4.8 - Parâmetros de desempenho da transmissão de dados (simulação para 50 leitos com 6 dispositivos cada).....	84
Tabela 4.9 - Parâmetros de desempenho da transmissão de dados (simulação para 60 leitos com 6 dispositivos cada).....	86
Tabela 4.10 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de token e com passagem de token (simulação para 10 leitos com 6 dispositivos cada).....	90
Tabela 4.11 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de token e com passagem de token (simulação para 25 leitos com 6 dispositivos cada).....	92


Tabela 4.12 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de token e com passagem de token (simulação para 50 leitos com 6 dispositivos cada). .....	94
Tabela 4.13 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de token e com passagem de token (simulação para 60 leitos com 6 dispositivos cada). .....	96

## LISTA DE ACRÔNIMOS

TDMA	Time Division Multiple Access
VTPE	Virtual Token-Passing Ethernet
FTT-Ethernet	Flexible Time-Triggered Ethernet
H-BEB	High Priority Exponential Backoff
PM-AH	Protocolo Multiciclos para Automação Hospitalar
PS	Provedor de Serviços
IP	Internet Protocol
IGMP	Internet Group Management Protocol
UDP	User Datagram Protocol
ECG	Eletrocardiograma
EEG	Eletroencefalograma
RMJ	Request Message Join
CMJ	Confirmation Message Join
MTP	Message Token Pass
DM	Data Message
IBWA	Indication Beginning Window Asynchronous
IAM	Indication Alarm Message
RRJ	Request Remove Join
RCM	Request Control Message
CCM	Confirmation Control Message
RAFM	Request Adjust Factor Message
RMN	Reindexing Message Node
ISM	Indication Synchronization Message
M/ID	Mestre/Identificador
ID	Identificador
SGBD	Sistema de Gerenciamento de Banco de Dados
DER	Diagrama Entidade-Relacional
1FN	1ª Forma Normal
2FN	2ª Forma Normal
3FN	3ª Forma Normal



RdP	Rede de Petri
NS	Network Simulator
CPU	Central Processing Unit
RAM	Random Access Memory
QoS	Qualidade de Serviço

## LISTA DE PUBLICAÇÕES

PERIÓDICOS NACIONAIS E INTERNACIONAIS		
Periódico	Título	Ano
 <p>REVISTA CIÊNCIA E TECNOLOGIA</p>	<p>Redes de Controle em Ambiente Hospitalar: Um protocolo multiciclos para automação hospitalar sobre IEEE 802.3 com IGMP Snooping</p>	2009

CONGRESSOS NACIONAIS E INTERNACIONAIS		
Congresso	Título	Ano
 <p>INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY</p>	<p>Hospital Networks: Multicycles Protocol for Hospital Automation over IEEE 802.3 with IGMP Snooping</p>	2009
 <p>INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS</p>	<p>Multicycles Protocol for Hospital Automation Based on Multicast Networking</p>	2008
 <p>34<sup>TH</sup> ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONIC SOCIETY</p>	<p>Medium Access Control: Multicycles Protocol for Hospital Automation over Multicast with IEEE 802.3</p>	2008



 <p>30<sup>TH</sup> ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY</p>	<p>Hospital Automation System RFID Based: Technology Embedded In Smart Cards (Cards, Tags and Bracelets)</p>	<p>2008</p>
 <p>CONGRESSO BRASILEIRO DE AUTOMÁTICA</p>	<p>RFID Aplicado a Automação Hospitalar: Desenvolvimento de um Sistema para Automação de Laboratórios de Análise Clínica</p>	<p>2008</p>

## 1. INTRODUÇÃO

Os avanços na área da eletrônica digital têm contribuído para o crescimento das aplicações distribuídas em que os dispositivos são dotados de certo poder de processamento (*smart devices*) (Dietrich e Sauter, 2000). Essa descentralização da computação é uma tendência dos sistemas de automação modernos (Pedreiras et. al., 2005). Nesse esquema, as funcionalidades do sistema estão presentes em vários elementos e não em apenas um, como descreve a computação centralizada (Valentim, 2008).

Nos processos de automação, existem alguns conceitos que são fundamentais tais como: aquisição de dados para controle de processos, monitoramento e processamento de sinais, redução de custos, otimização de processos etc. (Nitzan e Rosen, 2006).

Esses conceitos da automação já há algum tempo estão sendo incorporados em processos oriundos da automação hospitalar (Brooks e Brooks, 1998). Nesse sentido, surge uma série de requisitos pertinentes à área médica visando automatizar processos encontrados nesse tipo de ambiente (Valentim, 2008).

Em virtude do fato da comunicação ser um fator imperativo na automação hospitalar, em que, os sistemas geralmente são distribuídos, a rede para a transferência de dados passa a ser um fator fundamental. Isto porque, ela deve ser capaz de prover a troca de dados cumprindo as exigências do ambiente automatizado (Valentim, 2008).

Assim como as redes de escritório, as redes hospitalares preocupam-se fundamentalmente com os tempos médios de respostas das comunicações (Valentim, et. al., 2008), todavia não suportam os requisitos impostos aos sistemas de automação, tais como, as redes de automação industrial (Brito et. al., 2004), (Dolejs et. al., 2004) e (Carreiro et. al., 2005).

Esse problema decorre do fato das redes hospitalares basearem a sua comunicação no padrão *Ethernet* (IEEE 802.3), o qual é não-determinístico (Dolejs et. al., 2004). Isto inviabiliza a automatização de alguns processos que necessitam de confiabilidade e que impõe restrições temporais (Valentim, 2008).

As tecnologias de rede usadas em ambientes industriais aparecem como uma solução ao problema da automação hospitalar, porém mostram-se inviáveis pelo seu alto custo e baixa interoperabilidade (Carreiro et. al., 2005). Em contraposição, as redes *Ethernet* apresentam-se como padrão de alto desempenho, baixo custo e alta interoperabilidade (Dolejs et. al., 2004),

com isso, existem diversos projetos de pesquisa que procuram estabelecer um mecanismo que torne viável o uso dessas redes em ambientes hospitalares de modo a cumprir os requisitos impostos pelos mesmos. Nesse sentido os trabalhos sobre *Ethernet* para ambientes industriais resultaram numa série de protocolos que garantem o determinismo e a confiabilidade, como exemplos têm-se o TDMA (*Time Division Multiple Access*) de Brito et al., (2004), o VTPE (*Virtual Token-Passing Ethernet*) de Carreiro et. al. (2005), FTT-*Ethernet* (Pedreiras, 2005) e o H-BEB (Moraes e Vasques, 2005). Tais abordagens, embora cumpram requisitos temporais impostos pelos processos de automação, não se aplicam especificamente as redes hospitalares. Isso em virtude de terem sido desenvolvidos para os ambientes industriais e, portanto, não contemplarem aspectos pertinentes aos processos de automação hospitalar como a relação paciente leito, por exemplo (Valentim, 2008).

Nesse contexto, uma solução que atenda aos requisitos pertinentes a uma rede hospitalar é o PM-AH (Protocolo Multiciclos para Automação Hospitalar) (Valentim, 2008), efetivamente por se tratar de um protocolo determinístico destinado ao estabelecimento e manutenção de redes entre dispositivos biomédicos, de modo a otimizar o fator de utilização do meio de transmissão, e a garantir o cumprimento das metas temporais impostas pelos processos de monitoramento de pacientes (Valentim, 2008).

O modelo definido pelo PM-AH trabalha basicamente com três elementos: dispositivos biomédicos, leitos e Provedor de Serviços (PS).

Os dispositivos biomédicos são elementos que servem para efetuar medições de sinais vitais de pacientes ou realizar atividades de atuação sobre os mesmos. Como exemplo tem-se os sensores de ECG (Eletrocardiograma), de EEG (Eletroencefalograma), dos níveis de glicose no sangue dentre outros.

Os leitos da rede PM-AH alocam os pacientes e os dispositivos biomédicos a eles associados. Nesse ponto, um aspecto interessante do PM-AH é a segmentação da rede em leitos por meio do mapeamento destes em grupos *multicast* através do uso de um *switch* com suporte a IGMP *Snooping* (RFC 4541, 2006). Essa estratégia é interessante à medida que permite que a comunicação de cada leito da rede ocorra de modo independente, com isso, os dados referentes a um paciente não interferem nos dados de outro paciente em outro leito (Valentim et. al., 2008).

O Provedor de Serviços é a entidade responsável pela coordenação da rede como um todo. O PS é um dos elementos mais importantes da rede PM-AH, pois é responsável por inscrever e remover nós, organizar os nós dentro do leito, verificar a inatividade de dispositivos biomédicos, realizar testes de comunicação etc. (Souza, 2008). Para realizar essas

tarefas, o Provedor de Serviços deve estar inscrito em todos os grupos *multicast* (correspondentes aos leitos do ambiente hospitalar) de modo a receber toda mensagem que trafega pela rede.

A Figura 1.1 ilustra um exemplo de uma rede PM-AH contendo quatro leitos. Nessa Figura também é possível ver a segmentação da rede em grupos *multicast* (por meio do uso do *switch* com *IGMP Snooping*) e o Provedor de Serviços.

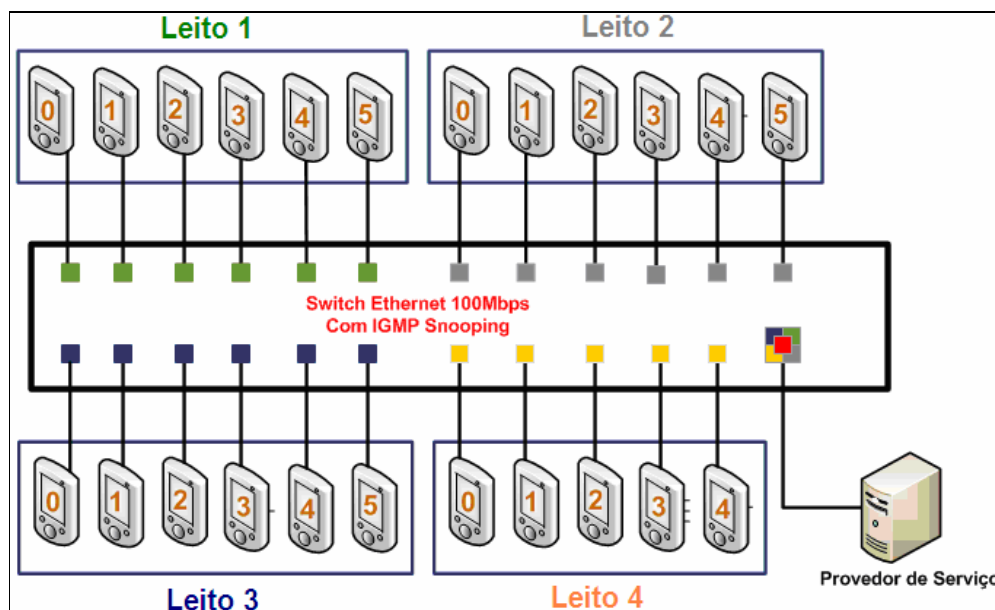


Figura 1.1 - Exemplo de uma Rede PM-AH.

Na Figura 1.1, é possível perceber que os dispositivos são referenciados dentro do leito por meio de índices. Sobre este aspecto, é importante destacar que a atribuição desses índices é feita pelo Provedor de Serviços no momento em que esses dispositivos solicitam sua inscrição no leito.

Com base no exposto e conforme o detalhamento a ser descrito nas seções subseqüentes desta dissertação, um dos elementos de maior importância em uma rede PM-AH é o Provedor de Serviços, sendo assim, o presente trabalho teve como objeto de estudo o desenvolvimento de modelos relacionados a esse elemento. Modelos estes especificados em três plataformas: Diagramas Entidade-Relacional de banco de dados, Redes de Petri e Network Simulator (NS versão 2). Ambas as plataformas com propósitos diferentes para avaliação do Provedor de Serviços PM-AH:

- A primeira com objetivo de avaliar aspectos pertinentes ao armazenamento de informações necessárias ao Provedor de Serviços no cumprimento de suas atividades;

- A segunda com objetivo de realizar a validação e a verificação formal do provedor de serviços PM-AH.
- A terceira com objetivo de observar o comportamento do Provedor de Serviços PM-AH quanto ao desempenho da rede;

### **1.1. Objetivo Geral**

Realizar análise de desempenho do Provedor de Serviços PM-AH, em cenários nos quais, a rede de automação hospitalar é submetida a sobrecarga. E neste ínterim, especificar, validar e verificar formalmente o Provedor de Serviços PM-AH em Redes de Petri e através de um modelo entidade e relacionamento baseado na álgebra relacional, o qual descreve o sistema de persistência de dados. Deste modo, provendo não apenas uma análise de desempenho, mas também formal, deste elemento essencial do protocolo PM-AH.

### **1.2. Objetivos Específicos**

Os objetivos específicos descrevem as metas a serem alcançadas. Para tanto, estão previstos as seguintes metas:

- Realizar pesquisa detalhada sobre o PM-AH;
- Levantar as atividades realizadas pelo Provedor de Serviços durante o processo de estabelecimento e manutenção da rede hospitalar;
- Modelar o banco de dados utilizado pelo Provedor de Serviços com vistas a realizar uma análise de desempenho que considere também o impacto gerado por este artefato;
- Modelar o Provedor de Serviços em Redes de Petri;
- Realizar a validação e a verificação formal do Provedor Serviços através do modelo especificado em Redes de Petri;
- Realizar análises de desempenho relativas ao tempo de processamento gasto nas operações de acesso ao banco na inscrição dos dispositivos biomédicos;
- Realizar análises de desempenho relativas à entrega de mensagens de dados dos dispositivos biomédicos para o Provedor de Serviços;

### **1.3. Organização do Trabalho**

Na seqüência, o texto abordado estrutura-se da seguinte forma:

- No Capítulo 2 apresenta-se a fundamentação teórica do trabalho, ou seja, o Protocolo Multiciclos para Automação Hospitalar;

- No Capítulo 3 apresenta-se a modelagem do banco de dados utilizado pelo Provedor de Serviços de uma rede PM-AH e da Rede de Petri referente às atividades realizadas pelo PS, bem como, a avaliação desses modelos;
- No Capítulo 4 apresenta-se uma série de análise de desempenho sobre diversas atividades realizadas pelo Provedor de Serviços;
- No Capítulo 5 têm-se as considerações finais do trabalho e os trabalhos em perspectiva;

## 2. FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo é abordado o referencial teórico utilizado na modelagem e na execução de análises de desempenho relativas ao Provedor de Serviços de uma rede PM-AH. Os tópicos aqui explorados dizem respeito ao Protocolo Multiciclos para Automação Hospitalar.

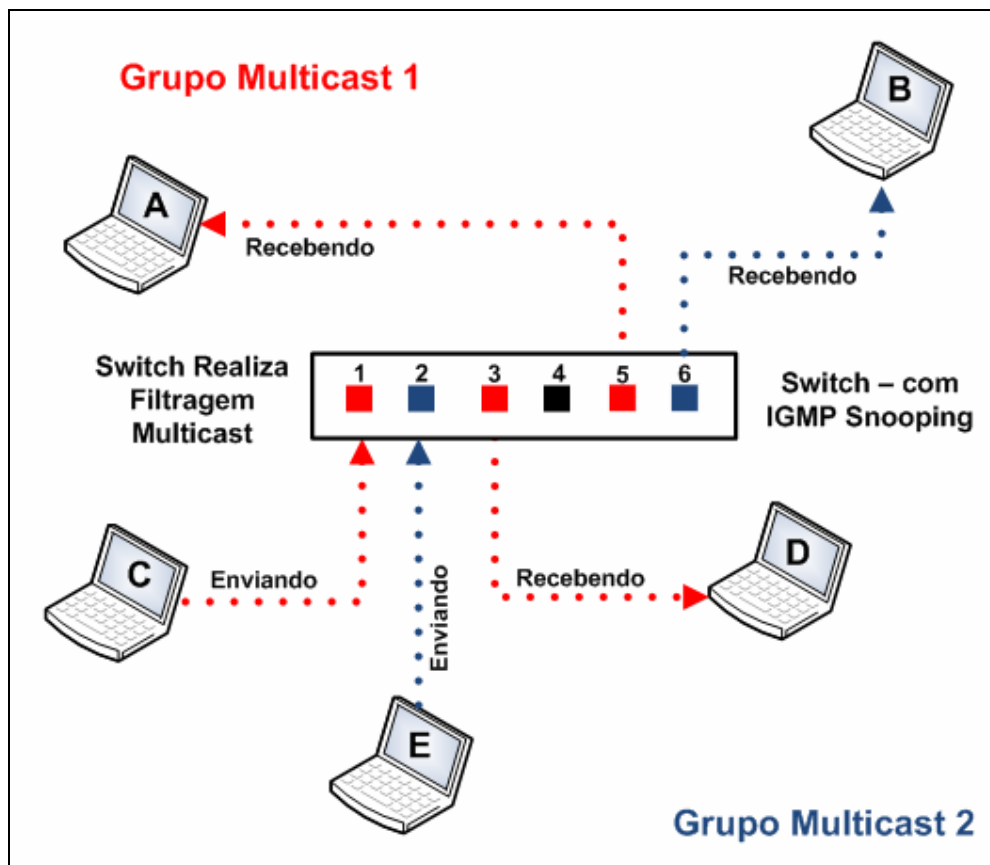
### 2.1. PM-AH: Protocolo Multiciclos Para Automação Hospitalar

O PM-AH é um protocolo destinado ao estabelecimento e manutenção de redes de comunicação entre dispositivos biomédicos (Valentim, 2008). Esse protocolo trabalha sobre a tecnologia *Ethernet*, em virtude do seu alto desempenho, baixo custo e expressiva interoperabilidade (Dolejs et. al., 2004), além do fato das redes de dados nos ambientes hospitalares serem todas baseadas nesse padrão, sendo assim, é possível reduzir os custos de instalação por meio do uso da infra-estrutura existente (Souza, 2008).

Um aspecto interessante do PM-AH é o uso do protocolo IP (*Internet Protocol*) (RFC 791, 1981) com endereçamento *multicast* de modo a segmentar a rede em leitos, em que, cada leito é visto como um grupo *multicast* e os dispositivos são vistos como nós dos grupos (Valentim, 2008).

Nesse aspecto, as mensagens originárias de um dispositivo biomédico são encaminhadas somente aos nós pertencentes ao mesmo leito em que esse dispositivo está associado, com isso, otimiza-se o uso da largura de banda disponível (Valentim et. al., 2008) e evita-se a interferência na troca de mensagens entre dispositivos pertencentes a leitos distintos (Souza, 2008).

A filtragem *multicast* realizada no PM-AH é feita por meio do uso de um *switch* com suporte à IGMP (*Internet Group Management Protocol*) *Snooping* (RFC 4541, 2006), o qual permite a filtragem de endereços *multicast* no nível de enlace. Nesse *switch*, o sistema de filtragem é definido quando um nó (dispositivo) se associa a um grupo *multicast*. Nesse momento, a porta a qual o dispositivo está conectado é vinculada ao grupo *multicast* ao qual o dispositivo se associou (Souza, 2008). A associação entre as portas do *switch* e grupos *multicast* é feita por meio de tabelas internas ao *switch* que são atualizadas à medida que as associações entre nós e grupos *multicast* acontecem. A Figura 2.1 ilustra um esquema de filtragem utilizando IGMP *Snooping*.



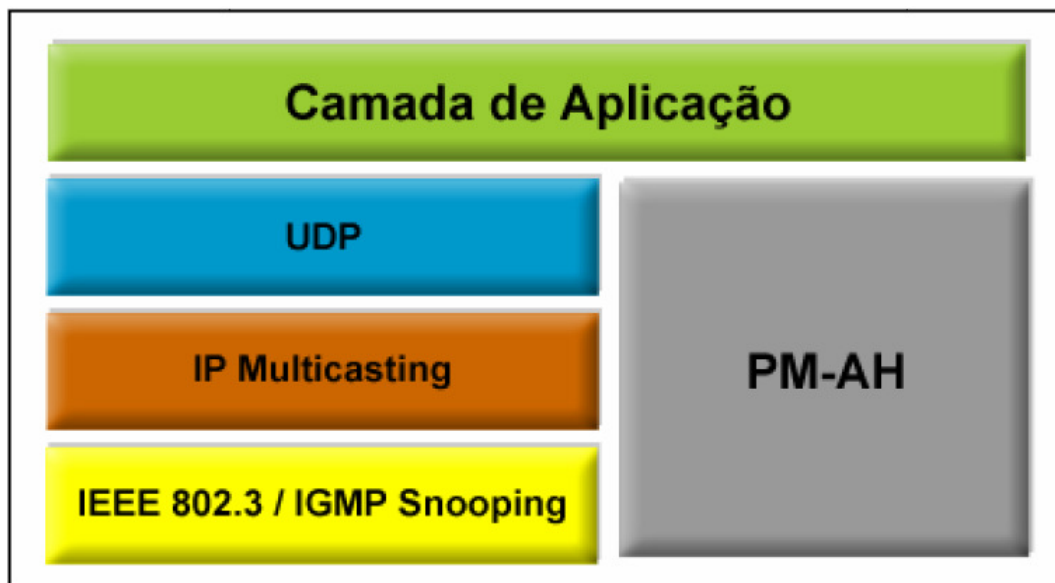
**Figura 2.1 - Esquema de filtragem de mensagens utilizando um switch com suporte a IGMP Snooping. (Fonte: Valentim, 2008)**

No esquema da Figura 2.1, existem dois grupos *multicast*: o grupo vermelho, cujos participantes são os nós A, C e D, e o grupo azul, cujos participantes são os nós B e E. Na situação demonstrada, o nó C e o nó E enviam mensagens no mesmo instante de tempo. Se o *switch* não suportasse *IGMP Snooping* o que aconteceria nesse exemplo é que, tanto a mensagem enviada pelo nó C como a enviada pelo nó E, seriam replicadas em todas as portas do *switch*, o que acarretaria em enfileiramentos e possíveis perdas de mensagens. Porém, como o *switch* fornece suporte a *IGMP Snooping* o que acontece é que as mensagens enviadas pelo nó C são replicadas apenas nas portas do *switch* que tem dispositivos pertencentes ao mesmo grupo *multicast* do nó C (no caso os nós A e D), assim como, a mensagem enviada pelo nó E é replicada apenas nas portas do *switch* que tem dispositivos pertencentes ao mesmo grupo *multicast* do nó E (no caso, o nó B). Dessa forma, não ocorre enfileiramento e os leitões podem operar paralelamente sem que um interfira no outro.

Por se tratar de aplicações com requisitos de tempo real (Valentim, 2008), o PM-AH trabalha na camada de transporte utilizando o UDP (*User Datagram Protocol*) (RFC 768,



1980). Sendo assim, a pilha de protocolos do PM-AH é tal como descrita na Figura 2.2 abaixo:



**Figura 2.2 - PM-AH na pilha de protocolos da Internet. (Fonte: Valentim, 2008)**

Sob o aspecto temporal, o PM-AH divide o tempo de comunicação no leito em ciclos temporais compostos por duas janelas denominadas janela síncrona e janela assíncrona (Valentim, 2008). O que diferencia uma janela síncrona de uma janela assíncrona é a forma com que os dispositivos se comunicam dentro delas.

Na janela síncrona, a comunicação é feita de forma coordenada segundo um modelo mestre-escravo, em que, cada leito possui um dispositivo atuando como mestre e os demais atuando como escravos. Nesse aspecto, o dispositivo mestre é responsável por coordenar o envio das mensagens provenientes dos nós do leito durante o transcorrer da janela (Valentim, 2008). Na janela assíncrona, a comunicação dentro do leito é feita de modo livre de coordenação, ou seja, os dispositivos são livres para se comunicar independentemente do mestre (Valentim, 2008).

Sob uma perspectiva global, a entidade responsável pela coordenação da rede como um todo é chamada Provedor de Serviços, o qual é designado a efetuar tarefas como a eleição do mestre de cada leito, inscrição e remoção de dispositivos, cadastro de alarmes e dados, realização de testes de comunicação, reindexação de nós etc. (Souza, 2008). A Figura 2.3 ilustra o esquema de uma rede PM-AH com todos os elementos descritos até agora:

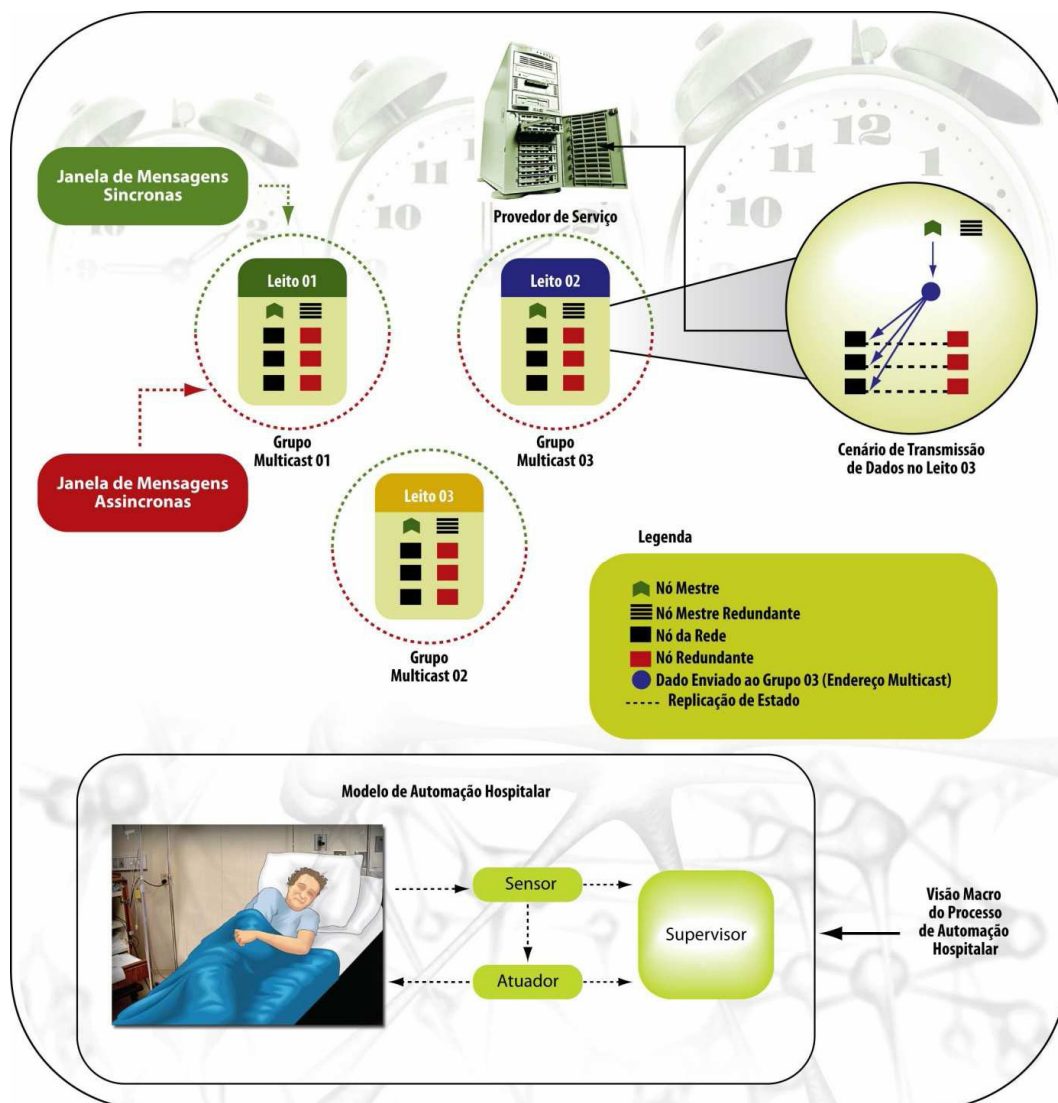


Figura 2.3 - Elementos de uma rede PM-AH (Fonte: Valentim, 2008).

## 2.2. Elementos do PM-AH

Basicamente, o PM-AH é composto por quatro elementos: dispositivos biomédicos (mestre e escravos), leitos hospitalares, ciclos temporais e Provedor de Serviços. Nesta sessão cada um desses elementos é abordado separadamente de modo a esclarecer o papel deles dentro da rede como um todo.

### 2.2.1. Dispositivos biomédicos e leitos hospitalares

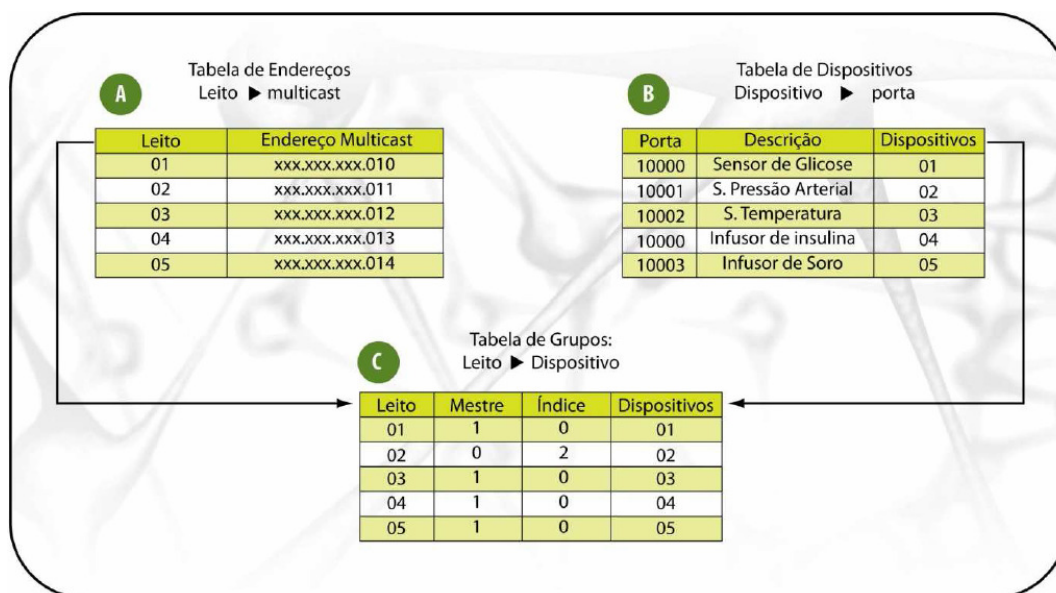
Os dispositivos biomédicos são os elementos da rede PM-AH responsáveis pela medição dos sinais vitais dos pacientes. Cada dispositivo da rede possui um código, indicativo

do tipo de dado que ele manipula, e uma porta pela qual são realizados o envio e recebimento de mensagens de dados (*Data Message – DM*) (Souza, 2008).

O código de um dispositivo deve ser um identificador único, por exemplo, o código referente a dispositivos de ECG (Entrocardiograma) é diferente do código referente a dispositivos de EEG (Eletroencefalograma) que é diferente do código de dispositivos de medição de glicose e assim por diante (Valentim et al., 2008).

Ao contrário do código, a porta associada a um dispositivo biomédico pode não ser única para cada tipo de dispositivo diferente. A justificativa para esse fato decorre da troca de mensagens de dados dentro do leito ser feita segundo um esquema de *publisher-subscribe* em que o dispositivo envia seus dados ao nós do grupo (leito) e esses decidem se efetuam o seu consumo ou não (Valentim, 2008). Desse modo, dispositivos que trabalham sobre o mesmo tipo de dados utilizam a mesma porta para efetuarem a publicação ou o consumo desses dados (Souza, 2008), por exemplo, dispositivos de medição de glicose e de infusão de insulina realizam a sua comunicação na mesma porta.

Assim como os dispositivos biomédicos, os leitos também possuem um código que serve para distingui-los dentro do ambiente hospitalar (Valentim et. al., 2008). Além desse código, cada leito possui também um endereço IP *multicast* associado, visto que, na rede PM-AH, cada leito é mapeado como sendo um grupo *multicast*. A associação entre dispositivos e leitos é tal como descrita na Figura 2.4 a seguir:



**Figura 2.4 – Esquema de associação de dispositivos em leitos. (Fonte: Valentim, 2008)**

A associação de um dispositivo em um leito da rede hospitalar ocorre por meio da troca de mensagens de solicitação e confirmação de inscrição entre os dispositivos

biomédicos e o Provedor de Serviços (Valentim, 2008). Nesse esquema, logo após ser iniciado, o dispositivo biomédico envia uma mensagem de requisição de inscrição (*Request Message Join* – RMJ) ao Provedor de Serviços (Valentim, 2008) indicando qual o leito no qual ele deseja se associar. Quando essa mensagem chega ao PS, o mesmo verifica se é possível efetuar a inscrição no leito desejado e, caso seja possível, envia uma mensagem de confirmação de inscrição (*Confirmation Message Join* – CMJ) ao dispositivo solicitante indicando o sucesso da sua inscrição e descrevendo o papel que esse dispositivo exercerá dentro do leito, o endereço *multicast* correspondente ao leito desejado, à porta onde o dispositivo realizará o envio ou consumo de dados e o índice pelo qual o dispositivo inscrito será referenciado dentro do leito (Valentim et. al., 2008).

As mensagens de solicitação e confirmação de inscrição são sempre enviadas em uma porta e grupo *multicast* específicos para essa finalidade, sendo assim, antes de se inscrever no grupo correspondente a um leito da rede PM-AH, o dispositivo deve estar inscrito nesses grupos *multicast* especiais para poder realizar as solicitações de inscrição em conjunto com o Provedor de Serviços (Valentim, 2008).

A inscrição do dispositivo num grupo *multicast* correspondente ao leito propriamente dito é feita somente após o recebimento da mensagem de confirmação de inscrição. Nesse esquema, quando o dispositivo recebe a mensagem de confirmação, ele desassocia-se dos grupos *multicast* especiais para a realização de inscrição de dispositivos e associa-se no grupo *multicast* indicado na mensagem de confirmação (Valentim et al, 2008).

O envio das mensagens de solicitação e confirmação de inscrição não interfere no processo de comunicação dentro do leito, uma vez que, o dispositivo solicitante só efetuará a sua inscrição no grupo *multicast* correspondente ao leito desejado, logo após ter recebido a mensagem de confirmação de inscrição enviada pelo Provedor de Serviços (Souza, 2008). Sendo assim, as ocorrências de solicitações de inscrição podem acontecer a qualquer momento independente da janela em que o leito desejado se encontra.

Em termos gerais, dentro de um leito (grupo *multicast*), um dispositivo pode exercer dois papéis: mestre ou escravo (Valentim, 2008).

Um dispositivo no papel de mestre é responsável pela coordenação do envio de mensagens de dados provenientes dos demais nós do leito e sinalização do início e término das janelas constituintes de um ciclo temporal por meio das mensagens de sincronização (*Indication Synchronization Message* - ISM) e início de janela assíncrona (*Indication Beginning Window Assynchronous* - IBWA) (Souza, 2008).

Para efetuar a tarefa de coordenação de envio de mensagens de dados, o mestre possui uma lista dos índices dos dispositivos que se encontram ativos no leito em um determinado instante de tempo (Souza, 2008). Por meio dessa lista, o mestre efetua a passagem de *token*, através de mensagens de passagem de *token* (*Message Token Pass – MTP*), durante a janela síncrona, dando vez a todos os dispositivos do leito para que eles efetuem a transferência de seus dados (Valentim, 2008).

Caso exista alguma remoção ou inscrição de dispositivo no leito, essa lista de índices deve ser atualizada como forma de manter o esquema de passagem de *token* o mais eficiente possível (Valentim, 2008). A atualização dessa lista é feita por meio de mensagens de reindexação (*Reindexing Message Node – RMN*) que são enviadas pelo Provedor de Serviços apenas durante as janelas assíncronas (Valentim, et al., 2008).

Os dispositivos escravos têm apenas a função de atuar ou medir sinais vitais dos pacientes (Valentim, 2008). Numa janela síncrona, esses dispositivos são elementos passivos, ou seja, só enviam mensagens caso recebam um *token* proveniente do mestre do leito. Numa janela assíncrona, esses dispositivos têm autonomia no envio de suas mensagens, ou seja, não precisam da autorização do mestre para efetuar o envio de alguma informação pela rede (Valentim, et. al., 2008).

Nas janelas assíncronas, não são efetuadas transferências de dados correspondentes a medidas de sinais vitais de pacientes. Sendo assim, a autonomia dos dispositivos escravos nessa janela limita-se a ações de envio de mensagens de alarme (*Indication Alarm Message – IAM*), realização de testes de comunicação e solicitação de remoção no leito (Valentim et al., 2008).

Um dispositivo é inscrito na condição de escravo quando o leito desejado já possui mestre, ou quando não possui, mas o dispositivo em questão não tem capacidade de processamento suficiente para realizar as tarefas pertinentes ao mestre (fator indicado num dos campos da mensagem de solicitação de inscrição) (Valentim, 2008).

Na atribuição de índices, o PM-AH estipula que o dispositivo mestre sempre possui o índice zero, enquanto os escravos podem ter um índice que se encontra numa faixa entre 1 e 127, totalizando assim, 128 possíveis inscrições dentro de um leito (Valentim, 2008).

### **2.2.2. Ciclos temporais: janelas síncronas e assíncronas**

O PM-AH é um protocolo que divide o tempo de comunicação no leito em ciclos temporais. Esses ciclos, por sua vez, são compostos por duas janelas de comunicação: janelas síncrona e assíncrona (Valentim, 2008).

Na janela síncrona, o objetivo principal é a transferência dos dados correspondentes às medições efetuadas pelos dispositivos biomédicos. Essa transferência é realizada segundo um modelo mestre-escravo em que o dispositivo eleito como o mestre do leito define qual dispositivo transmitirá seus dados em um determinado instante de tempo (Valentim, 2008).

A indicação que um dispositivo é detentor do direito de transmissão é feita através do envio de mensagens de passagem de *token* (Souza, 2008). Nesse esquema, o mestre envia essa mensagem a cada dispositivo inscrito no leito (incluindo o próprio mestre), um por vez, de forma que, ao recebê-la, o dispositivo envia uma mensagem de dados contendo informações relativas às medições dos sinais vitais efetuadas pelos dispositivos (Valentim, 2008).

Vale salientar que todas as mensagens de dados enviadas em todo e qualquer leito da rede, devem também ser enviadas ao Provedor de Serviços, para tanto, este deve estar inscrito em todos os leitos (grupos *multicast*) da rede de forma que possa receber toda e qualquer mensagem que trafegue por ela (Souza, 2008).

Na janela assíncrona, o comportamento não é coordenado como na janela síncrona, ou seja, se um dispositivo tem alguma mensagem para enviar, ele envia sem a necessidade da autorização do mestre (Valentim et al., 2008). O objetivo dessa janela é permitir a execução de tarefas que ocorrem de forma livre de sincronismo, tais como: notificação de alarmes, realização de testes de comunicação, reindexação e remoção de nós etc (Valentim et al., 2008).

A duração de um ciclo temporal é dada pela soma dos tempos necessários para a execução de uma janela síncrona ( $T_{js}$ ) e de uma janela assíncrona ( $T_{ja}$ ) (Valentim, 2008). A entidade responsável por calcular os tempos de duração de cada janela no PM-AH é o mestre do leito, nesse aspecto, adota-se que o tempo necessário para a execução de uma janela assíncrona é dado em função do tempo necessário para a execução de uma janela síncrona (Souza, 2008). Inicialmente, o tempo de cada janela é considerado igual.

As equações abaixo são utilizadas na definição dos tempos referentes à duração das janelas síncrona e assíncrona, bem como o tempo de duração de um ciclo completo. Em virtude dos mestres determinarem o tempo de duração de cada janela, as equações abaixo devem estar definidas em todo e qualquer dispositivo que tenha capacidade de processamento necessária para ser o mestre de um leito (Valentim, 2008).

$$T_{js} = \left( \sum_{m=1}^q T_{p_m} + T_v \right), \quad T_{pm} = \left( \frac{d}{c * \beta} \right), \quad T_v = \frac{L}{R} \quad \text{(Equação 2.1)}$$

Em que:

$q$ : Quantidade de nós inscritos no grupo (incluindo o nó mestre);

$T_{Pm}$ : Tempo de propagação do sinal no meio físico;

$T_V$ : Tempo de transmissão em função da vazão da rede;

$d$ : Distância que a mensagem vai percorrer;

$c$ : Velocidade da luz;

$\beta$ : Fator determinado em função do meio de transmissão;

$L$ : Comprimento da mensagem;

$R$ : Taxa de transmissão;

$$T_{ja} = T_{js} * \alpha \quad \text{(Equação 2.2)}$$

Em que:

$\alpha$ : Fator de ajuste temporal;

$$T_{ciclo} = T_{js} + T_{ja} \quad \text{(Equação 2.3)}$$

Em que:

$T_{ciclo}$ : Tempo de execução de um ciclo temporal;

Um ponto interessante nas equações acima diz respeito ao fator de ajuste temporal  $\alpha$  na Equação 2.2. Conforme dito, inicialmente o tempo de duração das janelas síncrona e assíncrona é considerado igual, nesse caso, esse fator é igual a um, porém, é possível reajustar o valor desse parâmetro por meio de mensagens de reajuste de fator temporal (*Request Adjust Factor Message* - RAFM). Essa mensagem pode ser enviada pelo Provedor de Serviços ao mestre de cada leito durante uma janela assíncrona e serve apenas para informar o novo valor de  $\alpha$  de modo a modificar o tempo de duração da janela assíncrona e, por conseguinte, do ciclo temporal.

### 2.2.3. Provedor de Serviços

O Provedor de Serviços é a entidade responsável pela coordenação da rede PM-AH como um todo (Valentim et. al., 2008). Dentre as tarefas cabíveis a ele, tem-se o cadastro de dados referentes às medidas efetuadas pelos dispositivos biomédicos, inscrição e remoção de dispositivos, atribuição e reindexação de índices, realização de testes de comunicação, cadastro de alarmes e ajuste do fator temporal determinante da duração das janelas assíncronas (Valentim, 2008).

Conforme dito, o Provedor de Serviços está inscrito em todos os leitos da rede, sendo assim, toda e qualquer mensagem enviada por um dispositivo biomédico será enviada também ao PS, possibilitando assim, que este realize todas as tarefas que são de sua responsabilidade (Souza, 2008).

Sob uma perspectiva comportamental, o Provedor de Serviços simplesmente aguarda a chegada de mensagens e realiza o processamento devido sobre as mesmas. Sendo assim, para um melhor entendimento das ações que o Provedor de Serviços deve tomar em cada situação é necessária a abordagem das diversas mensagens existentes no PM-AH. Dessa forma, na próxima seção é feita a abordagem dessas mensagens de modo detalhado visando sempre demonstrar a interação entre os dispositivos que as enviaram e o Provedor de Serviços.

### 2.3. Mensagens do PM-AH

Nessa seção são abordadas as mensagens utilizadas pelo PM-AH para o estabelecimento e manutenção das redes entre dispositivos biomédicos.

#### 2.3.1. Request Message Join (RMJ)

As mensagens de solicitação de inscrição (RMJ) são enviadas pelos dispositivos ao Provedor de Serviços logo após serem iniciados (Souza, 2008).

Ao enviar uma RMJ, o dispositivo solicitante espera durante cinco segundos uma mensagem de confirmação de inscrição. Caso não receba tal mensagem, o dispositivo reenvia a solicitação de inscrição repetindo esse processo por seis vezes (Valentim, 2008). O formato de uma RMJ é ilustrado na Figura 2.5 e seus campos são descritos na Tabela 2.1.

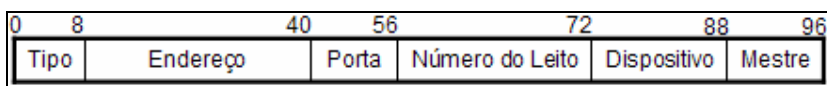


Figura 2.5 - Formato de uma RMJ.

Tabela 2.1 - Descrição dos campos de uma RMJ.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço <i>multicast</i> padrão para o envio de mensagens de solicitação de inscrição (apenas o Provedor de Serviços está inscrito nesse grupo).
Porta	16 bits	Porta padrão que indica o serviço relacionado às mensagens RMJ (este serviço é provido apenas pelo



		Provedor de Serviços).
Número do Leito	16 bits	Índice do leito ao qual o dispositivo deseja inscrever-se.
Dispositivo	16 bits	Código indicativo do tipo do dispositivo.
Mestre	8 bits	Campo indicativo da possibilidade do nó que está solicitando a inscrição exercer a função de mestre do leito. Nesse caso, se a informação desse campo for “11111111” então o dispositivo poderá ser mestre, caso não, se for “00000000” então o dispositivo terá que ser escravo. Nesse campo, apenas um <i>bit</i> seria necessário para indicar se um dispositivo pode ou não ser mestre do leito, a justificativa para o uso de 8 bits é a possibilidade de envio de outras informações durante o processo de requisição de inscrição.

### 2.3.2. Confirmation Message Join (CMJ)

As mensagens de confirmação de inscrição (CMJ) são enviadas pelo Provedor de Serviços como resposta a uma RMJ. Uma CMJ além de confirmar a inscrição de um dispositivo, também informa se este será mestre e qual o índice que tal dispositivo possuirá dentro do leito (Valentim et. al., 2008). O formato de uma CMJ é ilustrado na e Figura 2.6 seus campos são descritos na Tabela 2.2:

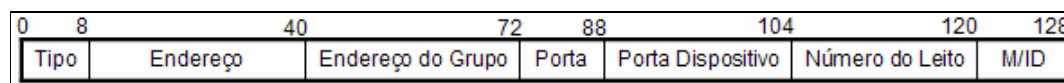


Figura 2.6 - Formato de uma CMJ.

Tabela 2.2 - Descrição dos campos de uma CMJ.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço <i>multicast</i> padrão referente ao serviço de confirmação de inscrição (apenas os dispositivos que não receberam uma CMJ devem estar associados a esse grupo).
Endereço do Grupo	32 bits	Endereço <i>multicast</i> referente ao leito em que o dispositivo se inscreveu.

Porta	16 bits	Porta padrão referente ao serviço de confirmação de inscrição (apenas dispositivos que não receberam uma CMJ devem consumir mensagens nessa porta).
Porta Dispositivo	16 bits	Porta que um dado tipo de dispositivo deve se associar para efetuar a publicação e consumo de dados.
Número do Leito	16 bits	Índice do leito onde o dispositivo se encontra.
M/ID	8 bits	Valor que informa se o dispositivo será mestre e o índice deste dentro do leito. O bit de mais alta ordem informa se o dispositivo atuará como mestre (0 – escravo, 1 – mestre), enquanto que os demais informam o índice pelo qual o dispositivo será referenciado.

### 2.3.3. Message Token Pass (MTP)

As mensagens de passagem de *token* são enviadas pelo mestre do leito aos dispositivos inscritos nesse e servem para coordenar o envio de mensagens de dados relativas às medições efetuadas pelos dispositivos biomédicos dentro de uma janela síncrona (Valentim, 2008).

Um aspecto importante na passagem de mensagens de *token* é o momento que o mestre deve enviá-las aos dispositivos do leito. A equação que descreve o intervalo de espera entre o envio de duas mensagens de *token* consecutivas é descrita abaixo (Valentim, 2008):

$$T_{pt} = (L_{mpt} + L_{md})/R + 2*AP \quad (\text{Equação 2.4})$$

Em que:

$T_{pt}$ : Intervalo de tempo para a passagem de *token*;

$L_{mpt}$ : Comprimento das mensagens de *token*;

$L_{md}$ : Comprimento das mensagens de dados;

$R$ : Taxa de transmissão (informada pelo Provedor de Serviços);

$AP$ : Atraso de propagação;

A Figura 2.7 ilustra o formato das mensagens de passagem de *token* e a Tabela 2.3 descreve os seus campos:

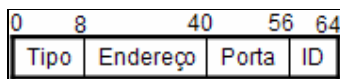


Figura 2.7 - Formato de uma MTP.

Tabela 2.3 - Descrição dos campos de uma MTP.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem
Endereço	32 bits	Endereço <i>multicast</i> referente ao leito que o dispositivo pertence.
Porta	16 bits	Porta onde os nós ficam aguardando as mensagens de <i>token</i> .
ID	8 bits	Campo de identificação do dispositivo que deve considerar a mensagem de <i>token</i> .

#### 2.3.4. Data Message (DM)

As mensagens de dados são enviadas pelos dispositivos biomédicos inscritos no leito logo após o recebimento de mensagens de *token* (Souza, 2008). A Figura 2.8 descreve o formato de uma DM enquanto que a Tabela 2.4 descreve os seus campos:

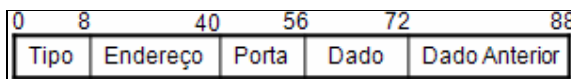


Figura 2.8 - Formato de uma DM.

Tabela 2.4 - Descrição dos campos de uma DM.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço <i>multicast</i> do grupo que vai receber as mensagens de dados.
Porta	16 bits	Porta que está relacionada ao tipo de dados que o dispositivo envia.
Dado	16 bits	Dado do dispositivo biomédico.
Dado Anterior	16 bits	Dado anterior (redundância temporal).

#### 2.3.5. Indication Beginning Window Assynchronous (IBWA)

As mensagens de início de janela assíncrona (IBWA) são enviadas pelo mestre do leito no fim de cada janela síncrona. Tais mensagens servem para notificar o término de uma

janela síncrona e início de uma assíncrona aos dispositivos inscritos no leito (Valentim et. al., 2008). O formato de uma IBWA é descrito na Figura 2.9 e a Tabela 2.5 descreve os seus campos.

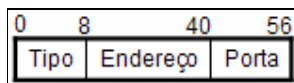


Figura 2.9 - Formato de uma IBWA

Tabela 2.5 - Descrição dos campos de uma IBWA.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço <i>multicast</i> do grupo cuja janela síncrona acabou de terminar.
Porta	16 bits	Porta padrão de recebimento de mensagens IBWA.

### 2.3.6. Indication Alarm Message (IAM)

As mensagens de alarme (IAM) são disparadas pelos dispositivos quando estes identificam alguma anomalia que esteja interferindo no processo de monitoramento do paciente (Valentim, 2008).

Vale salientar que essas mensagens são disparadas apenas nas janelas assíncronas como forma de não causar interferência no processo de troca de mensagens de dados realizado nas janelas síncronas (Souza, 2008). A Figura 2.10 ilustra o formato de uma IAM e a Tabela 2.6 descreve os seus campos.

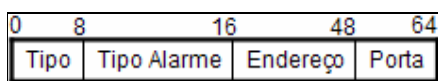


Figura 2.10 - Formato de uma IAM.

Tabela 2.6 - Descrição dos campos de uma IAM.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Tipo Alarme	8 bits	Código referente ao tipo do alarme.
Endereço	32 bits	Endereço <i>multicast</i> para o envio de mensagens de alarme (apenas o Provedor de Serviços deve estar associado a esse grupo).
Porta	16 bits	Porta padrão para receber mensagens de alarme.

### 2.3.7. Request Remove Join (RRJ)

As mensagens de remoção de dispositivos (RRJ) são enviadas em janelas assíncronas por dispositivos quando eles são desligados (Valentim, 2008).

Como essas mensagens são de natureza assíncrona, quando um dispositivo é desligado, o mesmo deve verificar se o leito ao qual ele está vinculado se encontra em uma janela assíncrona. Caso esteja, tal dispositivo deve enviar a mensagem de remoção, caso não, deve aguardar até o início da próxima janela assíncrona para então enviar a requisição de remoção (Souza, 2008). A Figura 2.11 ilustra o formato de uma RRJ e a Tabela 2.7 descreve os seus campos.

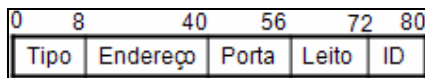


Figura 2.11 - Formato de uma RRJ.

Tabela 2.7 - Descrição dos campos de uma RRJ.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço <i>multicast</i> para o envio de mensagens de remoção (apenas o Provedor de Serviços deve estar associado a esse grupo).
Porta	16 bits	Porta onde o Provedor de Serviços fica aguardando mensagens de RRJ.
Leito	16 bits	Número do leito aonde o dispositivo se encontra.
ID	8 bits	Código de identificação do dispositivo.

### 2.3.8. Request Control Message (RCM) e Confirmation Control Message (CCM)

As mensagens de requisição (RCM) e confirmação de controle (CCM) são mensagens de teste de comunicação de dispositivos biomédicos visando verificar a atividade destes dentro do leito (Valentim, 2008).

Um teste de comunicação é realizado sempre que um dispositivo não envia os dados referentes ao monitoramento de seus pacientes por dois ciclos temporais consecutivos (Valentim et. al., 2008). Nesse caso, o Provedor de Serviços deve monitorar a comunicação de todos os dispositivos da rede e, caso um não se comunique por dois ciclos seguidos, enviar uma RCM ao dispositivo suspeito de inatividade (Souza, 2008).

Após enviar essa mensagem de teste de comunicação, o Provedor de Serviços deve esperar até o fim da próxima janela assíncrona por uma mensagem de confirmação (CCM) proveniente do dispositivo testado. Caso essa mensagem não seja recebida, então esse dispositivo é removido do leito (Valentim, 2008).

Tanto as mensagens de requisição como as de confirmação de controle, são enviadas apenas em janelas assíncronas. Dessa forma, a realização de testes de comunicação não interfere no processo de troca de dados entre os dispositivos biomédicos do leito (Souza, 2008). O formato da RCM e CCM é igual. A Figura 2.12 ilustra o formato de tais mensagens e a Tabela 2.8 descreve os seus campos.

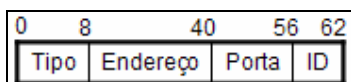


Figura 2.12 - Formato de uma RCM/CCM.

Tabela 2.8 - Descrição dos campos de uma RCM/CCM.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço <i>multicast</i> do grupo que vai receber uma RCM ou CCM.
Porta	16 bits	Porta padrão para o envio de mensagens de realização de testes de comunicação.
ID	8 bits	ID do dispositivo que vai ser testado.

### 2.3.9. Request Adjust Factor Message (RAFM)

A mensagem de ajuste de fator temporal (RAFM) é enviada pelo Provedor de Serviços ao mestre de um leito, apenas em janelas assíncronas, com o objetivo de ajustar o fator determinante do tamanho de uma janela assíncrona com base na Equação 2.2 (Valentim et. al., 2008).

A Figura 2.13 ilustra o formato de uma RAFM e a Tabela 2.9 descreve os campos dessa mensagem.

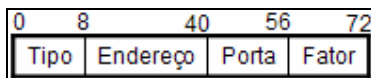


Figura 2.13 - Formato de uma RAFM.

Tabela 2.9 - Descrição dos campos de uma RAFM.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.

Endereço	32 bits	Endereço <i>multicast</i> do grupo que vai receber a mensagem de ajuste de fator temporal.
Porta	16 bits	Porta referente ao serviço de ajuste de fator temporal (apenas o mestre de cada leito deve estar habilitado a receber mensagens nessa porta).
Fator	16 bits	Valor que representa o fator de ajuste.

### 2.3.10. Reindexing Message Node (RMN)

As mensagens de reindexação de dispositivos (RMN) são enviadas pelo Provedor de Serviços ao mestre de cada leito como forma de notificar quais dispositivos encontram-se ativos neste em um determinado instante de tempo (Valentim, 2008).

As mensagens de reindexação são de natureza assíncrona e devem ser enviadas como forma de notificar a inscrição e remoções de dispositivos dentro do leito (Souza, 2008).

Uma RMN possui um campo chamado “Lista de Nós” de 128 bits, o qual corresponde a uma lista indicativa dos índices dos dispositivos ativos dentro do leito, dessa forma, se no leito estão ativos os dispositivos de índice 0, 1, 4, 9 e 120, por exemplo, nesse campo os bits nas posições 0, 1, 4, 9 e 120 devem ser ‘1’ e os demais devem ser ‘0’. A Figura 2.14 ilustra o formato de uma RMN. A Tabela 2.10 descreve os campos dessa mensagem:

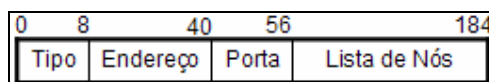


Figura 2.14 - Formato de uma RMN.

Tabela 2.10 - Descrição dos campos de uma RMN.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço <i>multicast</i> do grupo que vai receber a mensagem de reindexação.
Porta	16 bits	Porta onde o nó mestre vai receber a RMN (apenas o mestre deve estar habilitado a receber mensagens nessa porta).
Lista de Nós	128 bits	Informação relativa aos índices dos dispositivos que se encontram ativos no leito.

### 2.3.11. *Indication Synchronization Message (ISM)*

Uma mensagem de sincronização é enviada pelo mestre do leito aos demais dispositivos deste como forma de notificar o início de um novo ciclo temporal (Souza, 2008).

Uma ISM sincroniza os dispositivos os quais passam a esperar *tokens* logo após receberem uma mensagem de sincronização (Valentim, 2008). A Figura 2.15 ilustra o formato de uma ISM e a Tabela 2.11 descreve os seus campos:

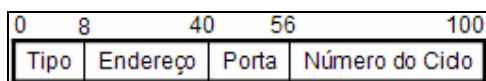


Figura 2.15 - Formato de uma ISM.

Tabela 2.11 - Descrição dos campos de uma ISM.

Campo	Tamanho	Descrição
Tipo	8 bits	Código referente ao tipo da mensagem.
Endereço	32 bits	Endereço do grupo <i>multicast</i> que receberá a mensagem de sincronização.
Porta	16 bits	Porta padrão para o recebimento das mensagens de sincronização.
Número do Ciclo	46 bits	Número atual do ciclo. Campo informativo do número do ciclo em que um leito se encontra.



### **3. MODELAGEM DO PROVEDOR DE SERVIÇOS DE UMA REDE PM-AH**

Esse capítulo descreve uma série de modelos relacionados às atividades executadas pelo Provedor de Serviços no armazenamento de dados e no processamento das mensagens descritas pelo PM-AH.

Dessa forma, na sessão 3.1 é ilustrado o modelo de entidades e relacionamento, que representa fisicamente o banco de dados utilizado pelo Provedor de Serviços e na sessão 3.2, é abordado um modelo comportamental do mesmo baseado em Redes de Petri.

#### **3.1. Modelagem formal do banco de dados para o Provedor de Serviços**

Em geral, todas as atividades realizadas pelo Provedor de Serviços estão associadas à manipulação de informações referentes aos dispositivos biomédicos e aos leitos da rede hospitalar (Valentim et. al., 2008). Sendo assim, é necessário que tais informações estejam armazenadas fisicamente, de modo a serem recuperadas sempre que for preciso e que não se percam caso haja algum problema no funcionamento da rede PM-AH – garantia de integridade dados. Nesse ponto surge, portanto, o conceito de banco de dados, que neste trabalho foi aplicado ao Provedor de Serviços utilizando as regras de normalização definidas por Boyce Code (Heuser, 1998).

Um banco de dados é um conjunto integrado de dados que tem por objetivo atender uma comunidade específica de usuários. Como em geral esses dados são compartilhados, o acesso a um banco pode ser de difícil gerenciamento, uma vez que, os arquivos contendo a descrição dos registros armazenados são bastante complexos e de difícil manipulação. Desse modo, uma ferramenta bastante utilizada por aplicações que envolvam o acesso a informações armazenadas em bancos de dados são os Sistemas de Gerenciamento de Banco de Dados (SGDB) (Date, 2004).

Um SGDB é um *software* que incorpora funções de definição, recuperação, alteração de dados armazenados, consulta, integridade referencial, atomicidade e controle de concorrência sobre bancos de dados. Deste modo, isentando, ou minimizando os desenvolvedores de implementarem tais tarefas em suas aplicações (Date, 2004). Dessa forma, o uso do SGDB é fundamental em determinadas aplicações, pois torna mais eficiente o trabalho dos projetistas de sistemas, visto que, a equipe de desenvolvimento não se

preocupam em implementar rotinas para gerenciar arquivos contendo o registro das informações armazenadas – esse aspecto torna as aplicações mais robustas, uma vez que, existem diversos SGBDs no mercado e na academia amplamente testados e validados. Dentre os SGBDs disponíveis no mercado, o mais comum é o relacional (Heuser, 1998), pois tem um forte embasamento matemático e também por ser um dos mais antigos a serem adotados pelo mercado de desenvolvimento de sistema, fator que permitiu ao longo do tempo diversos aprimoramentos. Com base neste aspecto, o sistema de armazenado de dados escolhido para implementar o módulo de persistência do Provedor Serviço neste trabalho foi relacional, sobre tudo por garantir a integridade referencial<sup>1</sup> dos dados.

Existem diversos modelos formais para descrever a estrutura de um banco de dados. Dentre esses modelos, os mais significativos são o modelo conceitual, o modelo lógico e o modelo físico (Heuser, 1998).

O modelo conceitual é um modelo abstrato que descreve a estrutura do banco independente do SGBD utilizado (Heuser, 1998). A técnica mais comum para ilustrar esse modelo é a abordagem entidade-relacional por meio dos diagramas entidade-relacional (DER). Esses diagramas contêm a descrição das entidades (elementos do sistema sobre os quais devem ser armazenadas informações), os relacionamentos (relações entre as entidades do banco) e os atributos (informações das entidades ou relacionamentos que devem ser armazenadas) (Heuser, 2008). A Figura 3.1 ilustra o DER do banco de dados para o Provedor de Serviços de uma rede PM-AH.

O DER da Figura 3.1 mostra que o banco de dados tem apenas duas entidades: “Dispositivos” e “Leitos”. A entidade “Dispositivos” é responsável pelo armazenamento dos tipos de dispositivos cadastrados no sistema, enquanto que entidade “Leitos”, armazena informações relativas aos leitos existentes na rede hospitalar. A Tabela 3.1 descreve os atributos da entidade “Dispositivos” e a Tabela 3.2 descreve os atributos da entidade “Leitos”.

Além das entidades descritas, o DER da Figura 3.1 possui três relacionamentos: “Associações”, “Dados” e “Alarmes”. O relacionamento “Associações” armazena as informações pertinentes à associação dos dispositivos biomédicos nos leitos da rede hospitalar, o relacionamento “Dados” armazena os dados referentes às medidas efetuadas pelos dispositivos biomédicos, e o relacionamento “Alarmes”, contém a descrição dos alarmes gerados ao longo de toda a comunicação da rede PM-AH.

A Tabela 3.3 descreve os atributos do relacionamento “Associações” enquanto que a

---

<sup>1</sup> É um conceito de banco de dados que assegura que as relações entre as tabelas permaneçam consistentes (Date, 2004).

Tabela 3.4 e a Tabela 3.5 descrevem os atributos dos relacionamentos “Dados” e “Alarmes” respectivamente.

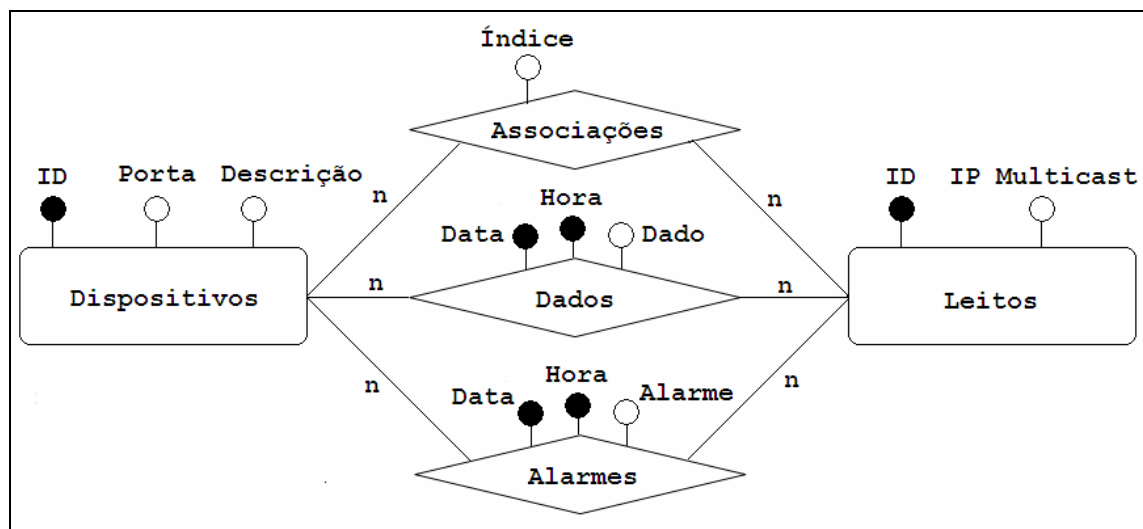


Figura 3.1 - DER do banco de dados para o Provedor de Serviços de uma rede PM-AH.

Tabela 3.1 - Descrição dos atributos da entidade "Dispositivos".

Atributo	Tipo do atributo	Descrição
ID	Chave primária	Identificador único para cada tipo de dispositivo cadastrado no sistema.
Porta	Atributo obrigatório	Número da porta onde o dispositivo realizará a publicação/consumo de dados;
Descrição	Atributo opcional	Breve descrição do dispositivo em questão.

Tabela 3.2 - Descrição dos atributos da entidade "Leitos".

Atributo	Tipo de atributo	Descrição
ID	Chave primária	Identificador único para cada leito cadastrado no sistema.
IP Multicast	Atributo obrigatório	IP Multicast referente ao leito em questão.

Tabela 3.3 - Descrição dos atributos do relacionamento "Associações".

Atributo	Tipo de atributo	Descrição
Índice	Atributo obrigatório	Índice atribuído pelo Provedor de Serviços no momento da inscrição do dispositivo no leito.

Tabela 3.4 - Descrição dos atributos do relacionamento "Dados".

Atributo	Tipo de atributo	Descrição
----------	------------------	-----------

Data	Chave primária	Data em que o dado foi cadastrado.
Hora	Chave primária	Hora em que o dado foi cadastrado.
Dado	Atributo obrigatório	Valor do dado cadastrado.

**Tabela 3.5 - Descrição dos atributos do relacionamento "Alarmes".**

Atributo	Tipo de atributo	Descrição
Data	Chave primária	Data em que o alarme foi cadastrado.
Hora	Chave primária	Hora em que o alarme foi cadastrado.
Alarme	Atributo obrigatório	Tipo do alarme cadastrado.

O modelo lógico é uma descrição do banco levando em conta aspectos relacionados ao SGBD utilizado (Heuser, 1998). Um SGDB relacional, implementa as entidades e relacionamentos por meio de tabelas e regras de projeto aplicadas ao problema a ser resolvido (Heuser, 1998). Sendo assim, a Figura 3.2 ilustra o modelo lógico para o banco de dados do Provedor de Serviços descrevendo as tabelas utilizadas em sua implementação.

```

Leitos(ID, IP Multicast)
Dispositivos(ID, Porta, Descrição)
Associações(ID Leito, ID Dispositivo, Índice)
    ID Leito referencia ID em Leitos
    ID Dispositivo referencia ID em Dispositivos
Dados(ID Leito, ID Dispositivo, Data, Hora, Dado)
    ID Leito referencia ID em Leitos
    ID Dispositivo referencia ID em Dispositivos
Alarmes(ID Leito, ID Dispositivo, Data, Hora, Dado);
    ID Leito referencia ID em Leitos
    ID Dispositivo referencia ID em Dispositivos

```

**Figura 3.2 - Modelo lógico para implementar o DER do banco de dados para o Provedor de Serviços da rede PM-AH usando um SGDB relacional.**

O modelo descrito na Figura 3.2 é composto por cinco tabelas: “Leitos”, “Dispositivos”, “Associações”, “Dados” e “Alarmes”. As tabelas “Dispositivos” e “Leitos” dizem respeito às entidades do DER da Figura 3.1. Sendo assim, as colunas dessas tabelas equivalem aos atributos das entidades descritas nesse diagrama.

As tabelas “Associações”, “Dados” e “Alarmes” dizem respeito aos relacionamentos do DER da Figura 3.1. Note que, comparando essas tabelas com os relacionamentos que elas

representam, existem duas colunas adicionais (“ID Leito” e “ID Dispositivo”) que não equivalem a nenhum atributo expresso em tais relacionamentos.

A presença dessas colunas é justificada pela necessidade de vincular cada registro das tabelas com um dispositivo e um leito cadastrados no sistema, visto que tais registros, implementam relacionamentos entre essas as entidades.

A solução adotada para realizar esses mapeamentos foi à inclusão das colunas “ID Leito” e “ID Dispositivo” na condição de chaves estrangeiras. Os valores referenciados por essas chaves indicam, respectivamente, o leito e os dispositivos referentes a cada registro da tabela, com isso, o número de colunas presentes nessas torna-se maior que o número de atributos expresso no DER da Figura 3.1.

O modelo físico é um modelo de baixo nível que retrata detalhes de armazenamento interno dos dados (Heuser, 1998). Esses detalhes de armazenamento não possuem nenhuma influência na programação de aplicações sobre o banco de dados, mas podem influenciar no desempenho no acesso ao mesmo (Heuser, 1998). Em virtude das suas características, esse modelo geralmente é utilizado apenas por profissionais que fazem sintonia do banco, sendo pouco utilizados no projeto de sistemas que envolvam o acesso as informações cadastradas nele. Desse modo, o projeto de banco de dados para o Provedor de Serviços não levou em conta aspectos relacionados ao modelo físico.

### **3.1.1. Verificação formal da modelagem de persistência de dados do Provedor de Serviços**

A normalização de um modelo de banco de dados baseia-se no conceito de forma normal. Uma forma normal é uma regra que deve ser obedecida por uma tabela para que ela seja considerada bem projetada (Heuser, 1998). Há diversas formas normais, ou seja, diversas regras, cada vez mais rígidas, para verificar tabelas relacionais (Heuser, 1998).

O processo de verificação formal do modelo esquemático (ver Figura 3.2) da base de dados proposta para o Provedor Serviços PM-AH foi realizada com base nas 1<sup>a</sup>, 2<sup>a</sup> e 3<sup>a</sup> formas normais propostas por Codd (1970). Para tanto, pode-se verificar que:

#### Primeira Forma Normal (1FN) de Codd (1970)

"Uma tabela está na 1FN, se e somente se, não possuir atributos multivalorado".

As tabelas Leitos, Dispositivos, Associações, Dados e Alarmes atendem está primeira regra, pois como pode ser visto no modelo esquemático apresentado na Figura 3.2, todos os

campos são atômicos, ou seja, não existe redundância em nenhuma tabela de campos em função de sua chave primária.

#### Segunda Forma Normal (2FN) de Codd (1970)

"Uma relação está na 2FN se, e somente se, estiver na 1FN e cada atributo não-chave for dependente da chave primária inteira, isto é, os atributos não podem depender parcialmente da chave".

Com relação a 2FN, com base na análise anterior, já foi verificado que todas as tabelas do modelo esquemático estão na 1FN, logo, a verificação foi remetida apenas a verificação da dependência parcial da chave primária. Neste caso, todos campos em suas respectivas tabelas dependem exclusivamente de suas chaves. Deste modo, garantindo que o modelo está efetivamente na 2FN.

#### Terceira Forma Normal (3FN) de Codd (1970)

"Uma relação está na 3NF, se ela estiver na 2NF e cada atributo não chave dessa relação não possui dependência transitiva, para cada chave candidata da relação."

Por fim, com base nas duas verificações anteriores (1FN e 2FN) para as relações do modelo esquemático da Figura 3.2, é possível afirmar então que modelo está 3FN, pois além de estar nas 1FN e 2FN como preconiza Codd (1970), as relações não apresentam nenhuma dependência transitiva, ou seja, todos os campos das tabelas dependem exclusivamente da chave (não há dependência indireta).

Esta normalização é fundamental, pois garante que o modelo de persistência do PS está livre de anomalias (de Inclusão, de Exclusão, ou de Alteração) e de redundância. Todavia, é importante observar as tabelas Dados e Alarmes (Figura 3.2) do modelo esquemático, pois ambas tem os mesmo campos, porém com valor semântico diferente para aplicação. Efetivamente, este é um processo de desnormalização que não gera nenhum dano ao sistema, quanto à consistência dos dados. O objetivo desta desnormalização foi aumentar a performance do sistema (escalabilidade) quando for necessário realizar consultas sobre esta base dados, pois irá mitigar problemas quando o banco estiver com um número elevado de tuplas e as consultas se tornarem mais custosas em termos de processamento.

Além da verificação formal mediante a análise do modelo da Figura 3.2, o presente texto aborda também uma análise da consistência dos dados presentes nas tabelas do modelo por meio da tentativa de realização de três operações ilegais sobre as mesmas: violação de chave primária, violação de chave secundária e remoção de um registro cuja chave primária é referenciada em outra tabela através de uma chave secundária.

A realização desses testes é importante, pois avalia se o modelo descrito na Figura 3.2 permite a execução de operações ilegais e, por conseqüente, violam o esquema de armazenamento de informações no banco de dados.

Na execução desses testes utilizou-se o banco de dados Postgre 8.3 com o auxílio do ambiente de gerenciamento pgAdmin III.

O primeiro teste realizado diz respeito à tentativa de violação da chave primária, ou seja, tentativa de inserção de dois registros com a mesma chave primária nas tabelas do banco. A realização desse teste é importante, pois a chave primária serve para diferenciar dois registros armazenados numa tabela, sendo assim, é de suma importância que o modelo de persistência do banco de dados não permita a inserção de dois registros com a mesma chave primária como forma de evitar ambigüidades nos registros armazenados (Heuser, 1998). A Figura 3.3 ilustra a tentativa de inserção de dois registros com a mesma chave primária nas Tabelas “Leitos” (item “a”) e “Dispositivos” (item “b”).

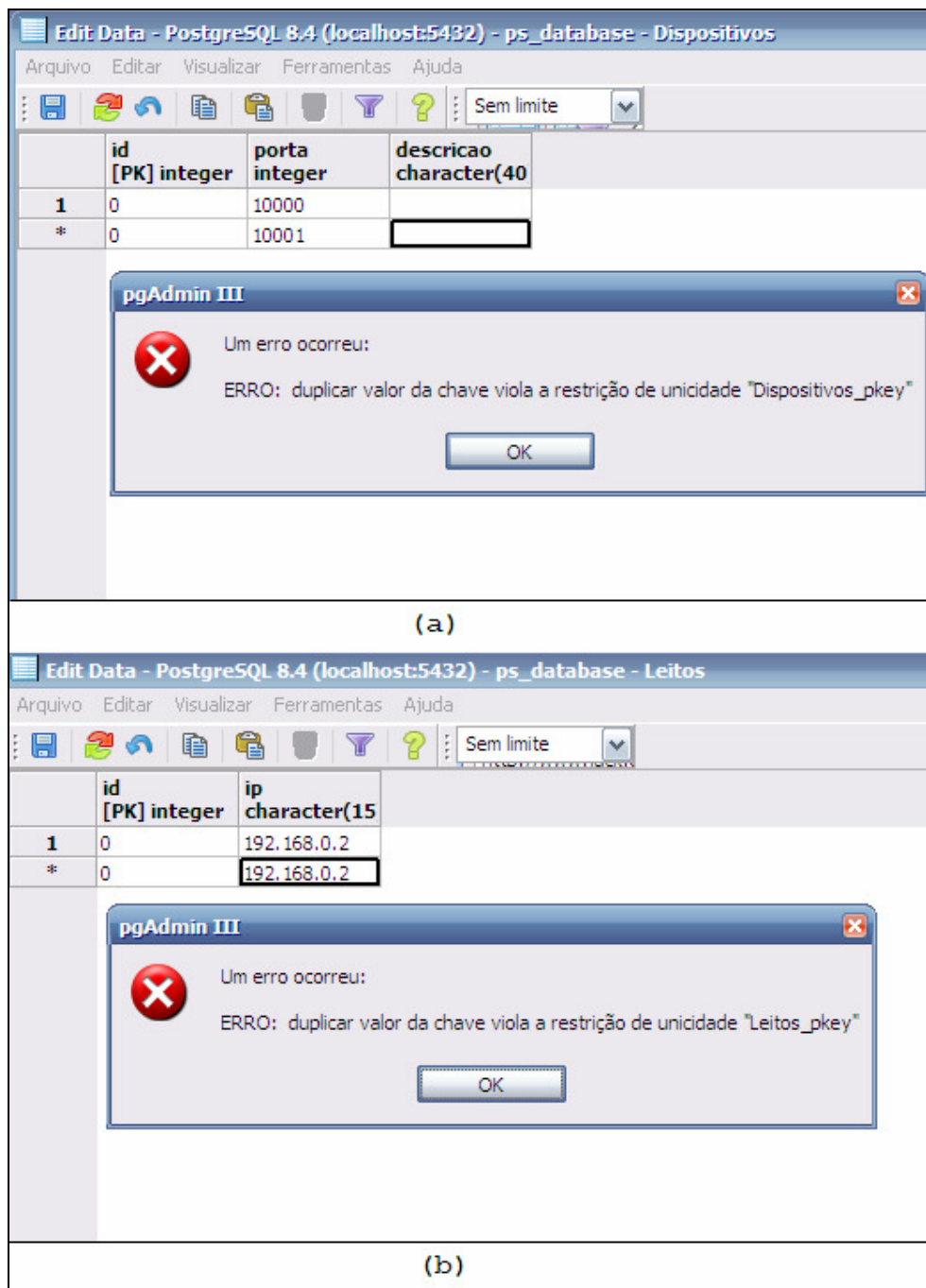


Figura 3.3 – Tentativa de inserção de chaves duplicadas. (a) Tabela “Leitos” (b) Tabela “Dispositivos”.

Os resultados descritos na Figura 3.3 apontam para a ocorrência de erros nas tentativas de inserção de registros com chaves primárias existentes tanto na Tabela “Leitos” quanto na Tabela “Dispositivos”. Dessa forma, o modelo de persistência de dados considerado é robusto quanto à violação de chaves primárias.

O próximo teste realizado diz respeito à tentativa de violação da chave secundária, ou seja, tentativa de inserção de registros cujo valor da chave secundária não corresponde a um



valor válido dentro dos registros que ela referencia. A execução desse teste é importante para verificar se o modelo de persistência de dados mantém a coerência nos registros armazenados em banco, visto que, ao armazenar uma chave secundária, deve-se ter a certeza que o valor da mesma corresponde a uma chave primária cadastrada na tabela referenciada (Heuser, 1998).

Para a execução desse teste considerou-se que as tabelas “Leitos” e “Associações” estavam dispostas tal como descrito na Figura 3.4. Sendo assim, a Figura 3.5 descreve o resultado da tentativa de violação da chave secundária nas tabelas “Associações”, sendo que, no item “a” tem-se o resultado do teste mediante a tentativa de violação da chave secundária que referencia o tipo de dispositivo cadastrado e no item “b” tem-se o resultado do teste da tentativa de violação da chave secundária que referencia o leito em que houve o cadastro. Vale destacar que a não realização de testes similares com as tabelas “Dados” e “Alarmes” é justificada pelo fato das mesmas implementarem o mesmo esquema de referência à chaves estrangeiras que a tabela “Associações”.

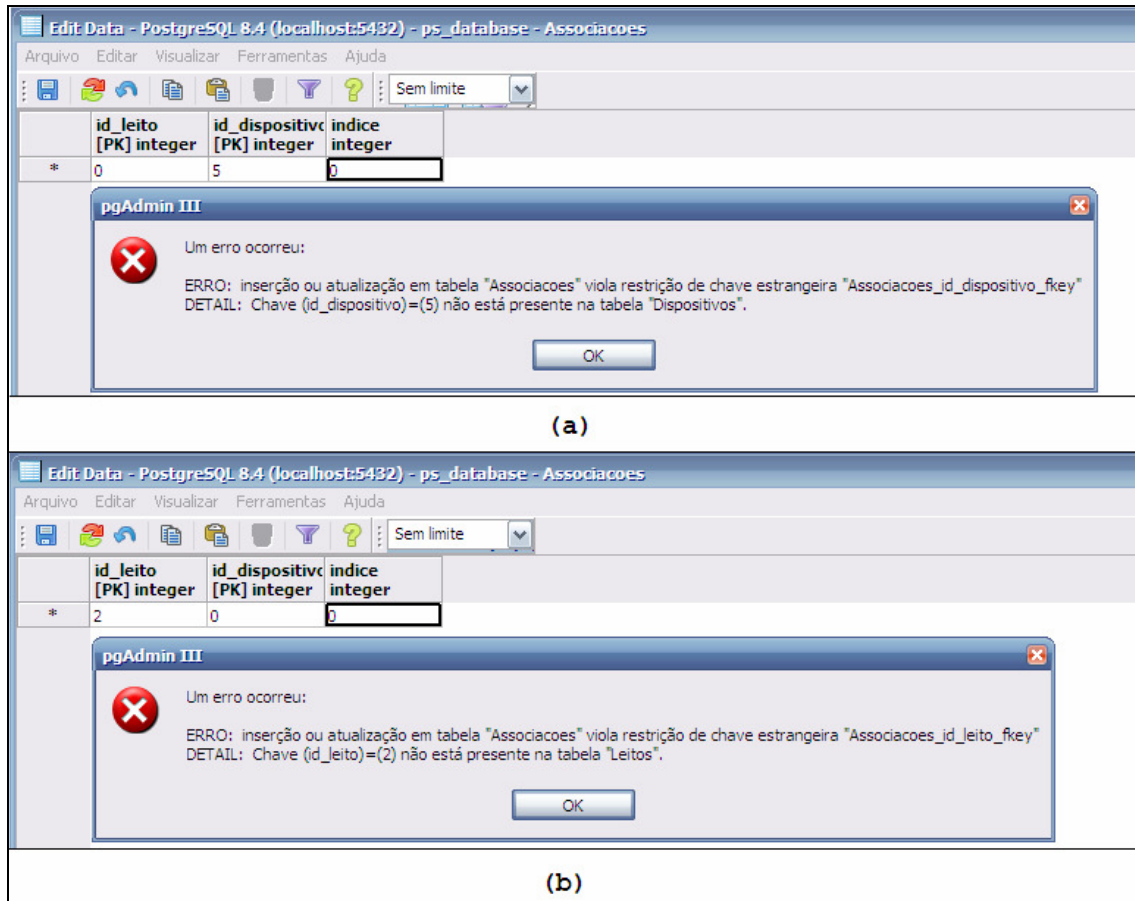
(a)

	id [PK] integer	porta integer	descricao character(40)
1	0	10000	
2	1	10001	
3	2	10002	
4	3	10000	
5	4	10003	
*			

(b)

	id [PK] integer	ip character(15)
1	0	192.168.0.2
2	1	192.168.0.3
*		

Figura 3.4 – Exemplo de preenchimento das tabelas “Dispositivos” e “Leitos” para realização de testes sobre o modelo de persistência de banco de dados. (a) Tabela “Dispositivos” (b) Tabela “Leitos”.



**Figura 3.5 – Resultados do teste de tentativa de violação de chave secundária. (a) Tentativa de violação da chave que referencia o dispositivo. (b) Tentativa de violação da chave que referencia o leito.**

Pelos resultados descritos acima, é perceptível que, tanto na tentativa de violação de chave estrangeira referente ao tipo do dispositivo como na tentativa de violação de chave estrangeira referente ao leito, não houve sucesso na realização das operações, com isso, o modelo de persistência de dados mostra-se também robusto a tentativas de violação de chaves estrangeiras.

O último teste realizado diz respeito à tentativa de remoção de registros cuja chave primária é referenciada em outra tabela através de uma chave secundária. A execução desse teste é importante, pois ao tentar remover um registro deve-se verificar se a chave primária deste é referenciada em outras tabelas, caso seja, a tentativa de remoção deve falhar como forma de manter a coerência dos dados armazenados nas tabelas (Heuser, 1998).

Para a realização desse teste foi considerado que a tabela “Associações” estava disposta tal como descrito na Figura 3.6. Sendo assim, a Figura 3.7 ilustra a tentativa de remoção de registros nas tabelas “Dispositivos” (item “a”) e “Leitos” (item “b”).

	id_leito [PK] integer	id_dispositivo [PK] integer	indice integer
1	0	0	0
2	0	1	1
3	0	2	2
4	1	3	0
5	1	4	1
*			

Figura 3.6 - Exemplo de preenchimento da tabela "Associações" para a realização dos testes.

(a)

(b)

Figura 3.7 - Resultados do teste de tentativa de remoção de registro nas tabelas cujas chaves primárias dos registros são referenciadas em outras tabelas. (a) Tentativa de remoção na tabela “Dispositivos” (b) Tentativa de remoção na tabela “Leitos”.

Pelos resultados descritos acima, é perceptível que, tanto na Tabela “Dispositivos” como na tabela “Leitos”, houve falha na tentativa de remoção de registros cujas chaves primárias são referenciadas em outras tabelas, sendo assim, o modelo de persistência de dados em questão foi capaz de manter a coerência das informações armazenadas em banco.

### 3.2. Modelagem do Provedor de Serviços em Redes de Petri

A Rede de Petri foi introduzida em 1966 por Carl Adam Petri (Petri, 1966) como uma técnica de modelagem capaz de representar sistemas paralelos, concorrentes, assíncronos e não-determinísticos (Maciel et. al., 1996) usando um forte modelo matemático.

As Redes de Petri podem ser utilizadas na análise e detecção de propriedades comportamentais de um sistema, tais como: presença ou ausência de *deadlocks* e *liveness*, limitação do modelo (Bonet et. al., 2007). Justamente por apresentar tais características que foi adotado a Rede de Petri como método para modelagem do Provedor de Serviços PM-AH.

Neste contexto, essa seção descreve o modelo em Rede de Petri referente ao Provedor de Serviços de uma rede PM-AH, para tanto, realizando análises das propriedades pertinentes a esse modelo.

A Figura 3.8 ilustra todo o modelo em Rede de Petri Provedor de Serviços PM-AH. Como forma de proporcionar um detalhamento dessa rede, bem como de facilitar a compreensão dos aspectos relacionados à mesma, dividiu-se a sua análise em quatro partes: na primeira parte foi analisada a Rede de Petri referente ao processamento de mensagens de requisição de inscrição (parte destacada em azul na Figura 3.8). Na segunda parte foi analisada a Rede de Petri referente ao processamento de mensagens trocadas dentro de uma janela síncrona (parte destacada em vermelho na Figura 3.8). Na terceira parte foi analisada a Rede de Petri referente ao processamento de mensagens trocadas dentro de uma janela assíncrona (parte destacada em verde na Figura 3.8) e na quarta parte foi realizada a análise das propriedades comportamentais do modelo completo do Provedor de Serviços.

Um aspecto relevante é que tanto a modelagem como a análise das Redes de Petri foram realizadas utilizando o software PIPE2 v2.5, uma ferramenta gráfica e de fácil uso que permite criar, salvar, carregar e analisar propriedades de Redes de Petri (Bonet et. al., 2007).

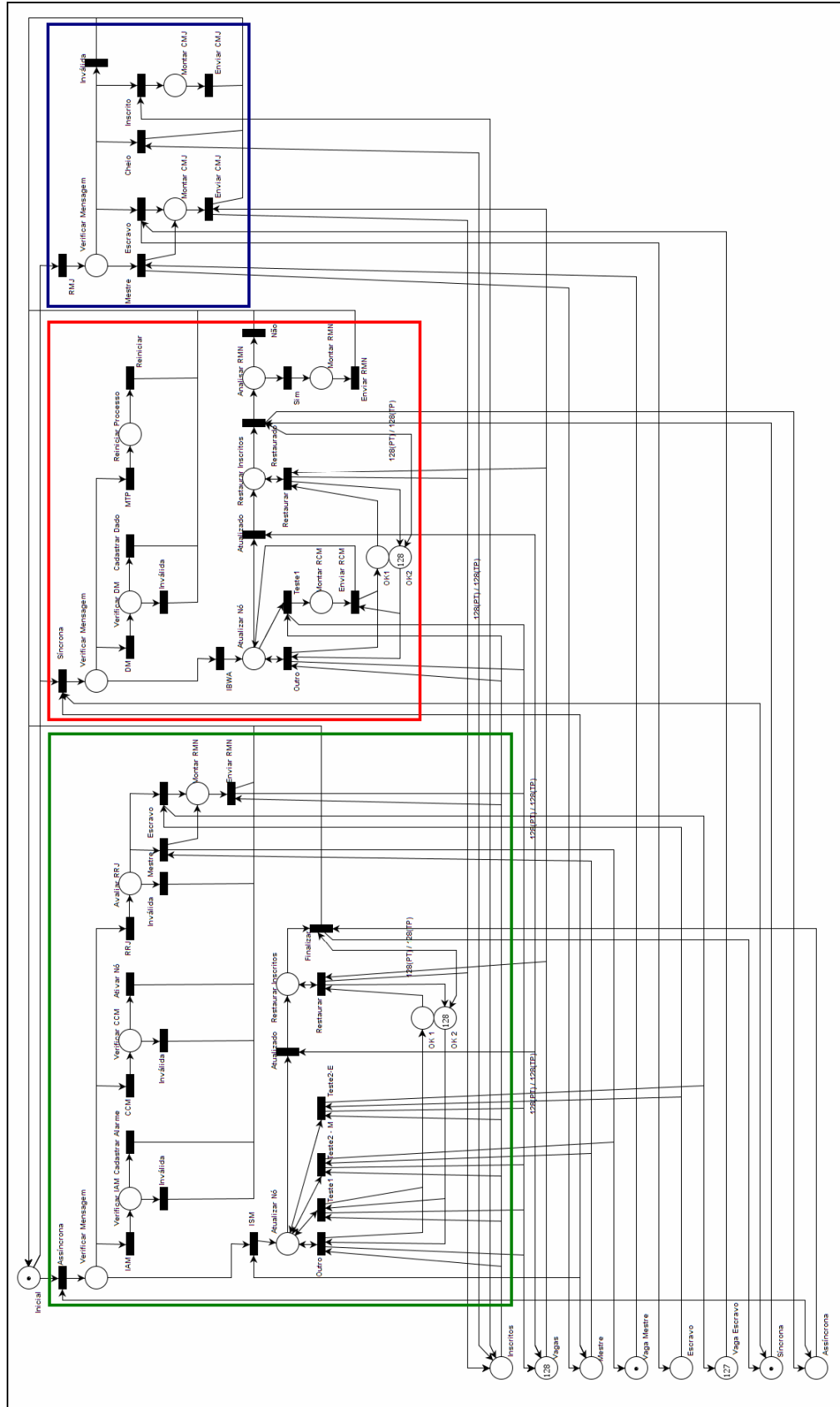


Figura 3.8 - Rede de Petri do Provedor de Serviços de uma Rede PM-AH.

### 3.2.1. Modelo em Rede de Petri: mensagens de requisição de inscrição

As mensagens de requisição de inscrição (*Request Message Join* – RMJ) são enviadas pelos dispositivos logo após serem iniciados. Ao receber essas mensagens, o Provedor de Serviços deve avaliar a requisição de inscrição e, dependendo do resultado dessa avaliação, enviar uma mensagem de confirmação (*Confirmation Message Join* – CMJ) ao dispositivo solicitante como forma de notificar o sucesso da sua inscrição no leito desejado (Valentim, 2008).

Em virtude do fato das mensagens de solicitação de inscrição serem enviadas antes de o dispositivo estar inscrito no leito (Valentim, 2008), o recebimento e processamento dessas são independentes do tipo da janela em que o leito referido se encontra no momento em que ocorre a solicitação.

A Figura 3.9 ilustra a Rede de Petri referente às atividades realizadas pelo Provedor de Serviços no recebimento de uma mensagem de requisição de inscrição.

Inicialmente o Provedor de Serviços encontra-se esperando a chegada de mensagens (esse fato é indicado na Rede de Petri acima por meio da presença de uma marcação no lugar “Inicial”). Ao receber uma requisição de inscrição, a transição “RMJ” é disparada e a marcação é transferida do lugar “Inicial” para o lugar “Verificar Mensagem”. Nesse ponto o Provedor de Serviços verifica se os dados contidos na requisição de inscrição são coerentes, ou seja, se o tipo do dispositivo e o índice do leito presentes na RMJ estão cadastrados no sistema.

Se os dados contidos na RMJ não forem coerentes, a transição “Inválida” é disparada devolvendo a marcação ao lugar “Inicial”, caso contrário, quatro transições são possíveis: “Mestre”, “Escravo”, “Cheio” e “Inscrito”.

Se o dispositivo puder ser mestre e o leito em questão não tiver mestre cadastrado então a transição “Mestre” é disparada. Nesse momento a marcação presente no lugar “Vaga Mestre” é transferida para o lugar “Mestre” indicando que o dispositivo solicitante será inscrito na condição de mestre do leito.

A transição “Escravo” é disparada se o dispositivo solicitante não tiver condição de ser mestre do leito ou se o leito já possuir um mestre. Quando a transição “Escravo” é disparada uma marcação do lugar “Vaga Escravo” é transferida para o lugar “Escravo” indicando que o dispositivo solicitante será inscrito na condição de escravo no leito.

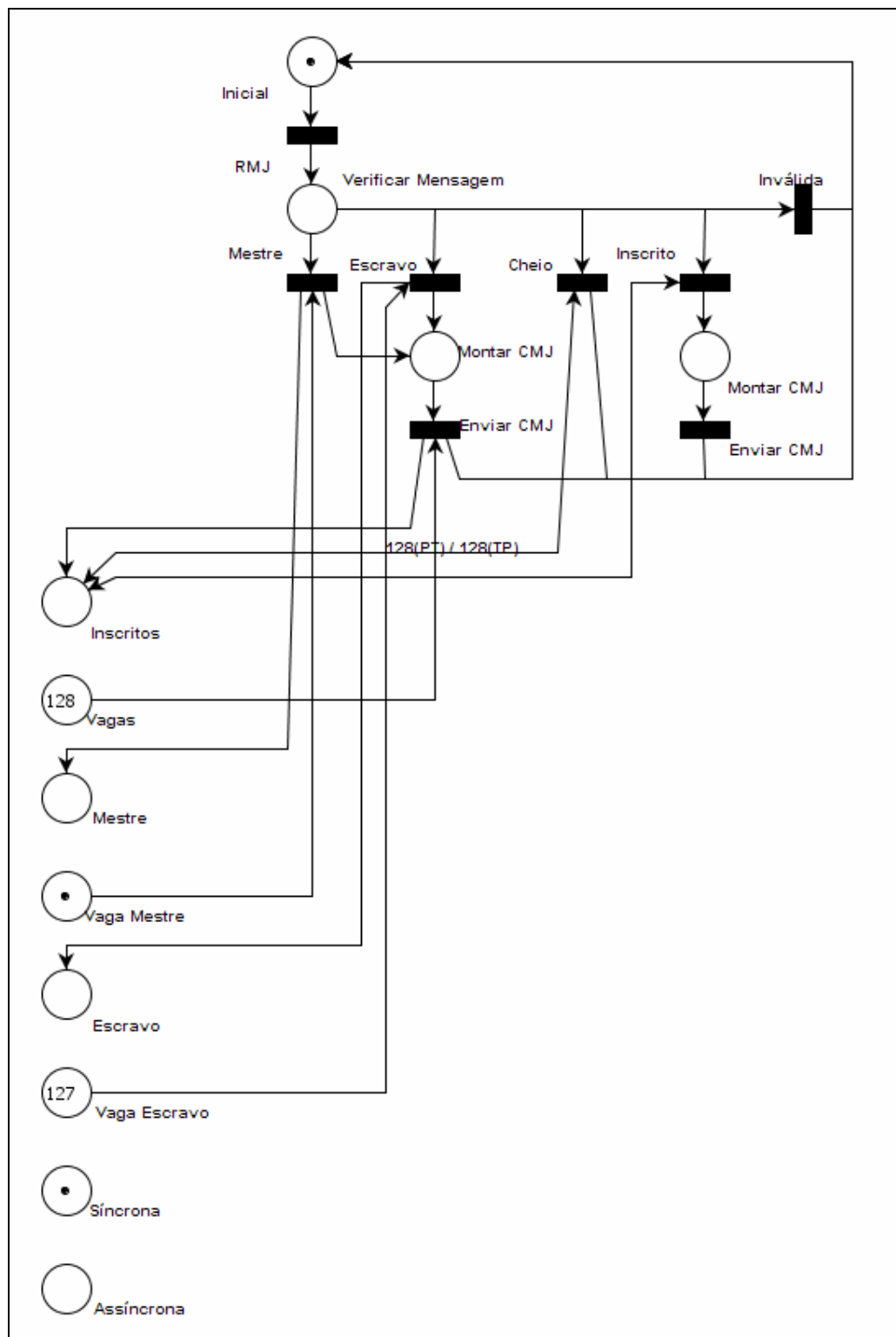


Figura 3.9 - Rede de Petri referente ao processamento de mensagens de solicitação de inscrição de dispositivos.

Tanto o disparo da transição “Mestre” como o da transição “Escravo” transferem a marcação de “Verificar Mensagem” para “Montar CMJ”. Nesse ponto o Provedor de Serviços monta a mensagem de confirmação de inscrição a ser enviada ao dispositivo solicitante e marca o mesmo como “Pendente” (inscrito, porém ainda não confirmado ao mestre do leito). Quando a mensagem está pronta, a transição “Enviar CMJ” é disparada indicando o envio da confirmação de inscrição. O disparo da transição “Enviar CMJ” atualiza o número de dispositivos inscritos no leito, por meio da transferência de uma marcação do lugar “Vagas” para o lugar “Inscritos”, e reinicia o processo de espera de mensagens através da transferência da marcação do lugar “Montar CMJ” para o lugar “Inicial”.

Se o dispositivo solicitante da inscrição já estiver inscrito no leito então a transição “Inscrito” é disparada. O disparo dessa transição move a marcação de “Verificar Mensagem” para “Montar CMJ”. Nesse momento o Provedor de Serviços monta uma mensagem de confirmação a ser enviada ao dispositivo que originou a solicitação de inscrição. O envio da mensagem de confirmação de inscrição a um dispositivo já inscrito no leito é justificado pelo fato que, ao enviar uma RMJ, o dispositivo espera cinco segundos pelo recebimento de uma CMJ, caso não receba tal confirmação, o mesmo reenvia à requisição de inscrição ao Provedor de Serviços (Valentim, 2008). Note que, diferente do caso em que houve a inscrição de mestre e escravos, à transição “Enviar CMJ” simplesmente indica o envio da mensagem de confirmação de inscrição no leito, ou seja, nenhuma nova inscrição é realizada. O disparo dessa transição também move a marcação de “Montar CMJ” para “Inicial” reiniciando o processo de espera de mensagens.

Se o dispositivo solicitante não estiver inscrito, mas o leito já possuir 128 inscrições realizadas, então não será possível atender a nova requisição, pois o leito já possui a quantidade máxima de dispositivos biomédicos permitidos pelo PM-AH (Valentim, 2008). Nesse caso, a transição “Cheio” é disparada transferindo a marcação de “Verificar Mensagem” para “Inicial”.

Um ponto interessante do modelo acima é que o processamento das mensagens de requisição de inscrição é independente da janela em que o leito desejado se encontra. Sendo assim, o sucesso na realização da inscrição de um dispositivo depende unicamente da coerência dos dados contidos na RMJ e do número de dispositivos inscritos no leito no momento em que ocorre a solicitação.



### 3.2.2. Modelo em Rede de Petri: mensagens trocadas em janelas síncronas

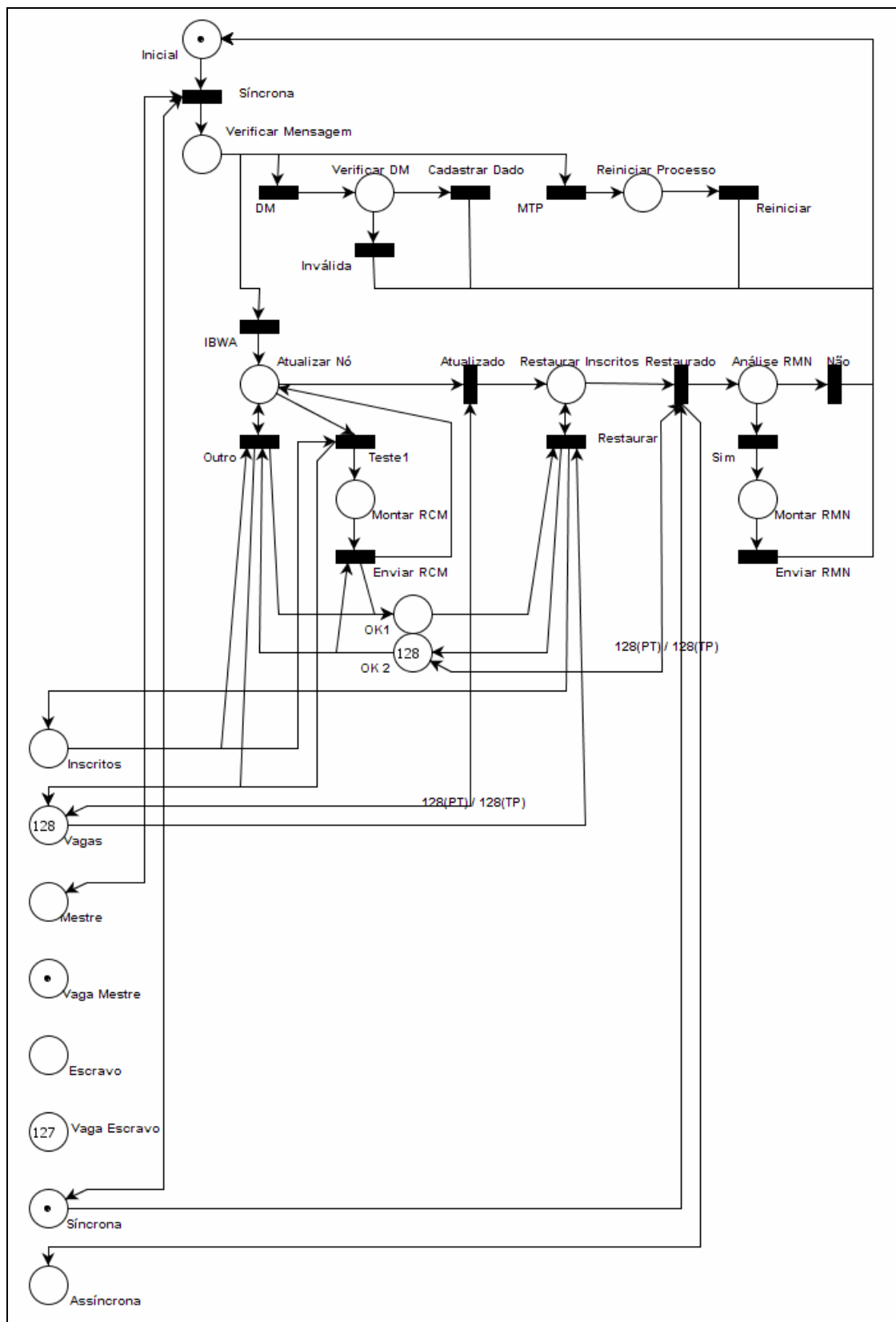
A janela síncrona é destinada ao envio coordenado, segundo um modelo mestre-escravo, dos dados referentes às medidas efetuadas pelos dispositivos biomédicos (Valentim, 2008).

Nessa janela, três mensagens podem ser recebidas pelo Provedor de Serviços: mensagens de passagem de *token* (*Message Token Pass* – MTP), mensagens de dados (*Data Message* – DM) e mensagens de início de janela assíncrona (*Indication Beginning Window Assynchronous* – IBWA) (Valentim, 2008). A Figura 3.10 ilustra a Rede de Petri referente ao processamento de mensagens trocadas dentro de uma janela síncrona.

Inicialmente o Provedor de Serviços encontra-se esperando a chegada de mensagens. Para que possa ocorrer à chegada de uma mensagem síncrona o leito deve estar dentro de uma janela síncrona e deve possuir um mestre associado. Tais condições são indicadas através da necessidade de marcações nos lugares “Mestre” e “Síncrona” para que a transição “Síncrona” esteja habilitada. A presença do mestre é necessária pelo fato de toda comunicação dentro de uma janela síncrona ser realizada segundo um modelo mestre-escravo (Valentim, 2008).

O disparo da transição “Síncrona” move a marcação do lugar “Inicial” para “Verificar Mensagem”. Nesse ponto o Provedor de Serviços analisa qual o tipo da mensagem síncrona recebida.

Se a mensagem recebida é uma mensagem de passagem de *token* então a transição “MTP” é disparada movendo a marcação de “Verificar Mensagem” para “Reiniciar Processo”. Nesse ponto o Provedor de Serviços não toma nenhuma atitude especial e a transição “Reiniciar” é disparada. O comportamento do Provedor de Serviços nessa ocasião é justificado pelo fato das mensagens de passagem de *token* servirem apenas para indicar qual dispositivo terá acesso ao meio físico (Valentim, 2008). Dessa forma, a única ação que poderia ser tomada no recebimento de tais mensagens é a indicação de qual dispositivo enviaria a próxima mensagem de dados forçando o Provedor de Serviços a avaliar somente as mensagens provenientes daquele dispositivo. A não adoção dessa atitude é justificada pela presença de enfileiramentos e perda de mensagens na porta do *switch* na qual o Provedor de Serviços está conectado (Souza, 2008). Sendo assim, a perda de uma mensagem de passagem de *token* comprometeria a análise da mensagem de dados proveniente do dispositivo detentor da posse do meio físico, uma vez que, a não chegada da mensagem de *token* acarretaria na não indicação que tal dispositivo era o detentor do direito de transmissão naquele instante de tempo.



**Figura 3.10 - Rede de Petri referente ao processamento de mensagens trocadas durante uma janela síncrona.**

Se a mensagem recebida é uma mensagem de dados então a transição DM é disparada movendo a marcação de “Verificar Mensagem” para “Verificar DM”. Nesse ponto, o Provedor de Serviços analisa se o dispositivo está inscrito no leito indicado na mensagem recebida e se ele já efetuou a transferência de dados dentro da janela síncrona atual. A análise para saber se o dispositivo em questão já realizou a transferência de dados é feita mediante a verificação do estado de tal dispositivo, o qual deve ser diferente de “Comunicante” (realizou comunicação dentro da janela síncrona atual) para que a mensagem em questão seja considerada. Essa análise é interessante pelo fato do Provedor de Serviços não realizar nenhum processamento especial sobre as mensagens de *token*, sendo assim, essa é uma forma de assegurar que um dispositivo só transfere seus dados uma vez dentro de cada janela síncrona. Se houver alguma irregularidade na análise da mensagem de dados então a transição “Inválida” é disparada movendo a marcação de “Verificar DM” para “Inicial”. Caso contrário, a transição “Cadastrar Dado” é disparada, indicando as ações de cadastro de dados, modificação do estado do dispositivo que enviou a mensagem para “Comunicante” e a transferência da marcação do lugar “Verificar DM” para “Inicial”.

Se a mensagem recebida for uma de início de janela assíncrona então a transição “IBWA” é disparada e a marcação é transferida de “Inicial” para “Atualizar Nó”. Nesse ponto o Provedor de Serviços atualiza o estado de cada dispositivo inscrito no leito. Nessa atualização, os dispositivos marcados como “Comunicante” são marcados como “Ativo” (comunicam-se regularmente), os marcados como “Ativo” são marcados como “Alerta” (estão um ciclo sem se comunicarem), os marcados como “Alerta” são marcados como “Teste1” (estão a dois ciclos sem se comunicarem), os marcados como “Pendente” (inscritos, porém ainda não confirmados ao mestre do leito) são marcados como “Ativo” e os marcados como “Teste1” e “Teste2” (possui apenas uma janela assíncrona para confirmar sua atividade dentro do leito) não sofrem alterações.

Vale salientar que ao modificar o estado de um dispositivo de “Pendente” para “Ativo” deve ocorrer à indicação da necessidade de envio de uma mensagem de reindexação (*Reindexing Message Node* – RMN). Essa mensagem é enviada ao mestre do leito e serve para atualizar a lista de índices dos dispositivos que estão ativos em um dado instante de tempo (Valentim, 2008).

Esse processo de atualização é justificado pelo fato do Provedor de Serviços marcar um dispositivo como “Comunicante” assim que recebe uma mensagem de dados proveniente do mesmo. Quando a janela síncrona termina, os dispositivos que não estão marcados como “Comunicante” foram os que não efetuaram comunicação. Dessa forma, os que antes estavam

marcados como “Ativo” não efetuaram transferência de dados na janela síncrona que acabou de terminar, logo estão a um ciclo temporal sem se comunicarem e com isso devem ser marcados como “Alerta”. Os que estavam marcados como “Alerta” não efetuaram transferência de dados na janela síncrona passada e também não efetuaram na janela síncrona que acabou de terminar, ou seja, estão a dois ciclos temporais sem efetuarem comunicação e com isso devem ser marcados como “Teste1”. A atualização referente aos estados “Teste1” e “Teste2” é realizada na janela assíncrona.

Vale salientar que o Provedor de Serviços verifica a cada atualização se houve a mudança do estado de um dispositivo de “Alerta” para “Teste1”. Em caso negativo a transição “Outro” é disparada mantendo a marcação em “Atualizar Nó”, em caso afirmativo a transição “Teste 1” é disparada movendo a marcação de “Atualizar Nó” para “Montar RCM”. Nesse ponto o Provedor de Serviços monta uma mensagem de teste de comunicação (*Request Control Message – RCM*) a ser enviada ao dispositivo que passou do estado de “Alerta” para “Teste 1”. Tal ação é justificada pelo fato do PM-AH estipular que dispositivos que estão a dois ciclos temporais sem se comunicar devem ser testados como forma de verificar a sua atividade dentro do leito (Valentim, 2008). Após montar a mensagem de teste, o Provedor de Serviços dispara a transição “Enviar RCM” indicando o envio de tal mensagem. O disparo dessa transição também move a marcação de “Montar RCM” para “Atualizar Nó”.

Tanto a transição “Outro” como a transição “Montar RCM” retiram uma marcação do lugar “Inscritos” e “OK 2” e adicionam uma marcação no lugar “Vagas” e “OK 1”. Tal procedimento é realizado como forma de assegurar que o Provedor de Serviços atualizará somente um número de dispositivos que equivale ao número de nós inscritos no leito. Vale salientar que os lugares “OK 1” e “OK 2” atuam como variáveis auxiliares que serão utilizadas para recuperar o número de nós inscritos antes do processo de atualização dos dispositivos.

Quando o Provedor de Serviços atualiza o estado de todos os dispositivos inscritos no leito, a transição “Atualizado” é habilitada e o seu disparo move a marcação de “Atualizar Nó” para “Restaurar Inscritos”. Nesse momento o Provedor de Serviços apenas restaura o número de marcações presentes no lugar “Inscritos” antes do processo de atualização de estado dos dispositivos. O disparo da transição “Restaurar” mantém a marcação no lugar “Restaurar Inscritos”, retira uma marcação dos lugares “Vagas” e “OK 1” e adiciona uma marcação nos lugares “Inscritos” e “OK 2”. Quando todas as marcações presentes no lugar “OK 1” forem retiradas, o número de nós inscritos antes do processo de atualização do estado dos dispositivos é restaurado e a transição “Restaurado” é habilitada. O disparo de tal

transição move a marcação presente no lugar “Síncrona” para o lugar “Assíncrona”, indicando a troca de janela, e transfere a marcação do lugar “Restaurar Inscritos” para o lugar “Análise RMN”. Nesse ponto, o Provedor de Serviços verifica se existe alguma indicação de mensagem de reindexação pendente. Caso não exista a transição “Não” é acionada movendo a marcação de “Análise RMN” para “Inicial”, caso exista a transição “Sim” é acionada, movendo a marcação de “Análise RMN” para “Montar RMN”. Nesse ponto o Provedor de Serviços monta a mensagem de reindexação a ser enviada ao mestre do leito. O disparo da transição “Enviar RMN” indica envio de tal mensagem e move a marcação de “Montar RMN” para “Inicial”.

Um ponto interessante no modelo acima é que após o processamento das mensagens de início de janela assíncrona, a transição “Síncrona” passa a estar desabilitada. Essa propriedade é interessante, pois evita o processamento de mensagens de natureza síncrona durante o transcorrer de uma janela assíncrona.

### **3.2.3. Modelo em Rede de Petri: mensagens trocadas em janelas assíncronas**

A janela assíncrona é destinada à troca de mensagens entre os dispositivos biomédicos e o Provedor de Serviços de forma não coordenada (Valentim, 2008).

Nessa janela, quatro mensagens podem ser recebidas pelo Provedor de Serviços: mensagens de notificação de alarmes (*Indication Alarm Message – IAM*), confirmação de teste de comunicação (*Confirmation Control Message – CCM*), solicitação de remoção de nós (*Request Remove Join – RRJ*) e de sincronização (*Indication Synchronization Message – ISM*) (Valentim, 2008). A Figura 3.11 ilustra a Rede de Petri referente ao processamento de mensagens trocadas durante uma janela assíncrona.

Inicialmente o Provedor de Serviços encontra-se esperando a chegada de mensagens. Para que possa ocorrer à chegada de uma mensagem assíncrona o leito deve estar dentro de uma janela assíncrona e deve possuir pelo menos um dispositivo inscrito. Tais condições são indicadas através da necessidade de marcações nos lugares “Inscritos” e “Assíncrona” para que a transição “Assíncrona” esteja habilitada. Se a mensagem recebida é uma de notificação de alarme então a transição “IAM” é disparada e a marcação é transferida de “Verificar Mensagem” para “Verificar IAM”. Nesse ponto, o Provedor de Serviços verifica se o dispositivo que gerou o alarme está inscrito no leito indicado na mensagem. Se o dispositivo não estiver inscrito então a transição “Inválida” é disparada, caso contrário, a transição

“Cadastrar Alarme” é disparada indicando o cadastro do alarme descrito na mensagem. Tanto o disparo da transição “Cadastrar Alarme” como da transição “Inválida” transferem a marcação do lugar “Verificar IAM” para “Inicial”.

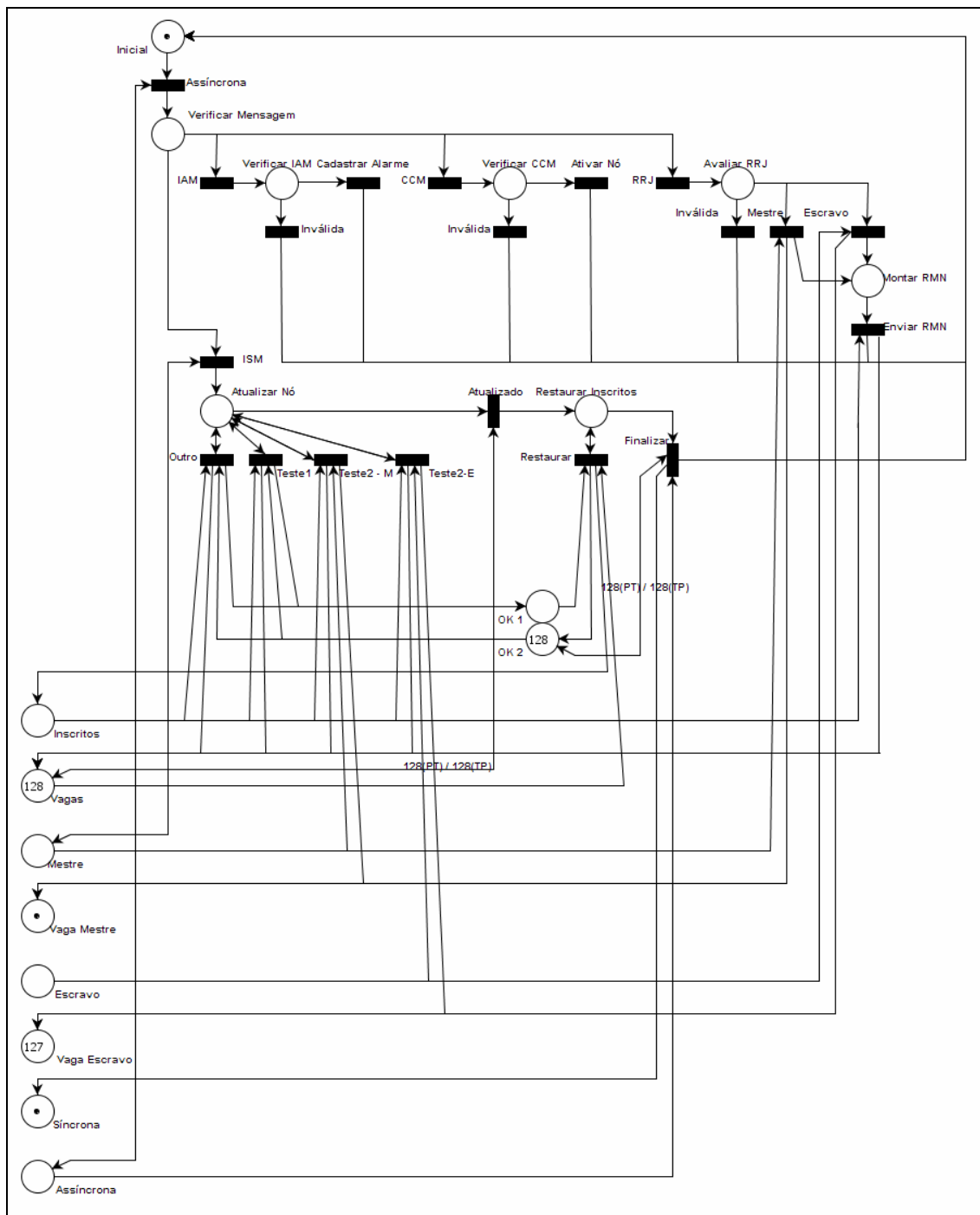


Figura 3.11 - Rede de Petri referente ao processamento de mensagens trocadas durante uma janela assíncrona.

Quando a transição “Assíncrona” é disparada a marcação é transferida do lugar “Inicial” para “Verificar Mensagem”. Nesse ponto o Provedor de Serviços analisa o tipo da mensagem assíncrona recebida.

Caso a mensagem recebida seja uma mensagem de confirmação de teste de comunicação, a transição “CCM” é disparada transferindo a marcação de “Verificar Mensagem” para “Verificar CCM”. Nesse ponto, o Provedor de Serviços verifica se o dispositivo que enviou a confirmação de teste de comunicação está inscrito no leito indicado na mensagem e se esse dispositivo encontra-se em teste (seu estado é “Teste 1” ou “Teste 2”). Se o dispositivo não estiver inscrito ou não estiver em teste então a transição “Inválida” é disparada, caso contrário, a transição “Ativar Nó” é disparada indicando que o Provedor de Serviços marcou o nó que enviou a mensagem como “Ativo”. Vale salientar que tanto a transição “Ativar Nó” como a transição “Inválida” transferem a marcação de “Verificar CCM” para “Inicial”.

Se a mensagem recebida for uma mensagem de requisição de remoção de dispositivo, então a transição “RRJ” é disparada e a marcação é transferida de “Verificar Mensagem” para “Verificar RRJ”. Nesse ponto, o Provedor de Serviços verifica se o dispositivo que enviou a requisição de remoção está inscrito no leito indicado na mensagem. Se o dispositivo não estiver inscrito, então a transição “Inválida” é disparada e a marcação é movida de “Verificar RRJ” para “Inicial”, caso contrário, duas situações são possíveis: o dispositivo é mestre ou escravo do leito. Se o dispositivo que originou a mensagem de remoção for mestre então a transição “Mestre” é disparada e com isso uma marcação é transferida dos lugares “Inscritos” e “Mestre” para os lugares “Vagas” e “Vaga Mestre”, respectivamente. Se o dispositivo que originou a mensagem é escravo então a transição “Escravo” é disparada e com isso uma marcação é transferida dos lugares “Inscritos” e “Escravos” para os lugares “Vagas” e “Vaga Escravo”, respectivamente. Vale salientar que, tanto no disparo das transições “Mestre” e “Escravo”, a marcação é movida do lugar “Verificar RRJ” para “Montar RMN”. Nesse ponto o Provedor de Serviços monta uma mensagem de reindexação a ser enviada ao mestre do leito em que houve as remoções. O disparo da transição “Enviar RMN” indica o envio da mensagem de reindexação e transfere uma marcação de “Inscritos” para “Vagas” sinalizando assim a remoção do dispositivo. Além disso, a transição “Enviar RMN” também move a marcação de “Montar RMN” para “Inicial”.

Se a mensagem recebida for uma de sincronização então a transição ISM é disparada e a marcação é transferida de “Verificar Mensagem” para “Atualizar Nó”. Note que o disparo da transição “ISM” depende da existência de um mestre inscrito no leito. Isso é justificado

pelo fato de tais mensagens serem enviadas apenas pelo mestre do leito no fim de cada janela assíncrona (Valentim, 2008).

No lugar “Atualizar Nó” o Provedor de Serviços analisa o estado de cada dispositivo inscrito no leito e toma as medidas adequadas para cada caso. Nesse ponto quatro transições podem ser disparadas: “Outro”, “Teste 1”, “Teste 2 - M” e “Teste 2 - E”.

Se o dispositivo analisado for “Pendente”, “Ativo” ou “Alerta” então a transição “Outro” é disparada indicando que não houve nenhuma mudança no estado desse dispositivo.

Se o estado do dispositivo analisado for “Teste 1” então a transição “Teste 1” é disparada. O disparo dessa transição atualiza o estado do dispositivo analisado para “Teste 2”. Essa atualização é justificada pelo fato dos dispositivos no estado “Teste 1” terem sido testados no início da janela assíncrona atual e como não chegou nenhuma mensagem de confirmação de teste de comunicação proveniente do dispositivo testado durante o transcorrer da janela assíncrona então esse dispositivo terá apenas mais uma janela assíncrona para confirmar a sua atividade no leito (Valentim, 2008), logo o seu estado deve ser modificado para “Teste 2”.

Tanto a transição “Outro” como a transição “Teste 1” retiram uma marcação do lugar “Inscritos” e “OK 2” e adicionam uma marcação no lugar “Vagas” e “OK 1”. Tal procedimento é realizado como forma de assegurar a restituição do número de nós inscritos antes do processo de atualização de dispositivos uma vez que os lugares “OK 1” e “OK 2” atuam como variáveis auxiliares utilizadas na recuperação das marcações presentes nos lugares “Inscritos” e “Vagas”.

Se o estado do dispositivo for “Teste 2” então duas transições podem ser disparadas: “Teste 2 - M” e “Teste 2 - E”. Se o dispositivo avaliado for mestre então a transição “Teste 2 - M” é disparada indicando a remoção do mestre do leito. Se o dispositivo avaliado for escravo então a transição “Teste 2 - E” é disparada indicando a remoção de um dispositivo escravo do leito. A remoção de um dispositivo que está no estado “Teste 2” é justificada pelo fato que foi realizado um teste de comunicação com esse dispositivo e a resposta desse teste não foi recebida na janela assíncrona passada (quando o estado do dispositivo era “Teste 1”) e nem na janela assíncrona que acabou de terminar. Como um dispositivo testado tem apenas duas janelas assíncronas para responder ao teste de comunicação (Valentim, 2008), os dispositivos em “Teste 2” devem ser removidos no fim da janela assíncrona, pois o seu prazo para responder ao teste de comunicação foi esgotado.

Nesse processo de remoção, o disparo da transição “Teste 2 - M” indica a remoção do mestre do leito por meio da transferência de uma marcação dos lugares “Inscritos” e “Mestre”



para os lugares “Vagas” e “Vaga Mestre”, enquanto que o disparo da transição “Teste 2 – E” indica a remoção de um escravo do leito por meio da transferência de uma marcação dos lugares “Inscritos” e “Escravos” para os lugares “Vagas” e “Vaga Escravo”. Um ponto interessante é que, tanto no disparo de “Teste 2 – M” quanto no disparo de “Teste 2 – E”, não houve alteração nas marcações presentes nos lugares “OK 1” e “OK 2”. Como esses lugares atuam como variáveis auxiliares para a recuperação do número de nós inscritos antes do processo de atualização de estado dos dispositivos, a não alteração na marcação presente nesses lugares implica na não restituição dos dispositivos que foram removidos no disparo das transições “Teste 2 – M” e “Teste 2 – E”. Vale salientar que ao remover um dispositivo que se encontra no estado “Teste 2” (disparo das transições “Teste 2 – M” e “Teste 2 – E”) deve haver a indicação da necessidade de envio de uma mensagem de reindexação para o mestre do leito.

Quando o Provedor de Serviços atualiza o estado de todos os dispositivos inscritos no leito, a transição “Atualizado” é habilitada e o seu disparo move a marcação de “Atualizar Nó” para “Restaurar Inscritos”. Nesse momento o Provedor de Serviços apenas restaura o número de marcações presentes no lugar “Inscritos” antes do processo de atualização de estado dos dispositivos do leito. O disparo da transição “Restaurar” mantém a marcação no lugar “Restaurar Inscritos”, retira uma marcação dos lugares “Vagas” e “OK 1” e adiciona uma marcação nos lugares “Inscritos” e “OK 2”. Quando todas as marcações presentes no lugar “OK 1” forem retiradas, o número de nós inscritos é restaurado e a transição “Finalizar” é habilitada. O disparo dessa transição move a marcação presente no lugar “Assíncrona” para o lugar “Síncrona”, indicando a troca de janela, e transfere a marcação do lugar “Restaurar Inscritos” para o lugar “Inicial”.

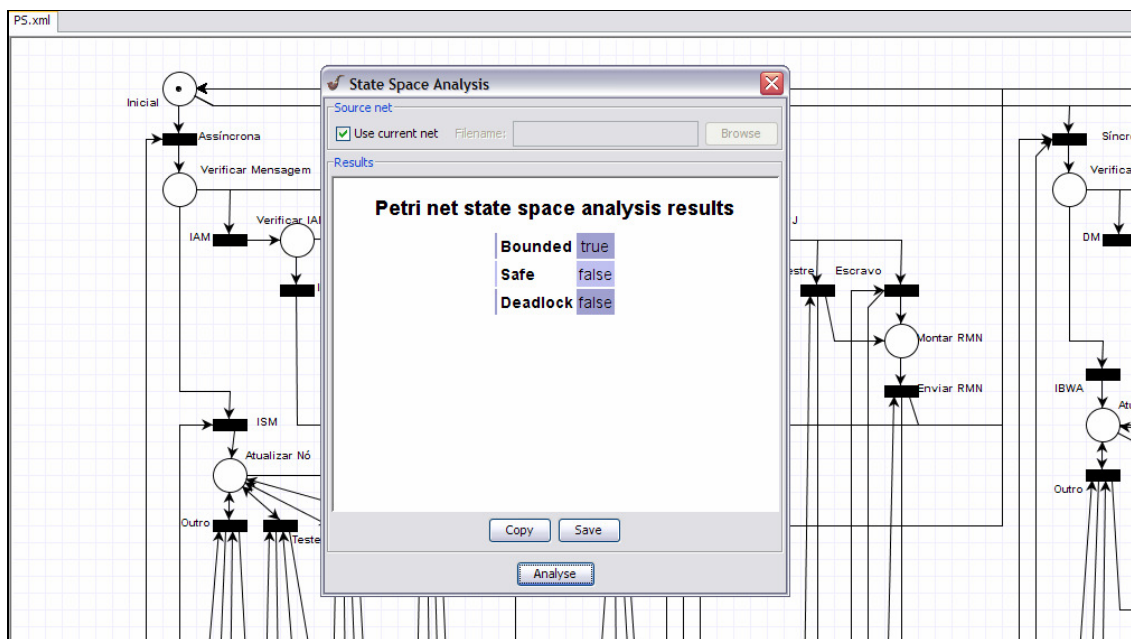
Desse modo, o modelo descrito para a janela assíncrona complementa as atividades realizadas na janela síncrona. Vale salientar que, em virtude da não existência de um mecanismo para ajuste de fator temporal, o modelo acima não prevê o envio de mensagens que alterem o tempo de duração de uma janela assíncrona em função do tempo de duração de uma janela síncrona (*Request Adjust Factor Message - RAFM*). Dessa forma, o tamanho das janelas é sempre igual (fator de ajuste temporal igual a um).

#### **3.2.4. Análise das propriedades da Rede de Petri do Provedor de Serviços**

Nessa sessão é realizada uma análise das propriedades comportamentais do modelo em Rede de Petri do Provedor de Serviços de uma rede PM-AH apresentado na Figura 3.8. A

análise dessas propriedades é importante, pois serve para validar o modelo proposto e assim permitir a sua implementação.

A Figura 3.12 mostra a verificação das propriedades comportamentais do modelo em questão. Nessa análise, as propriedades verificadas são: limitação do modelo, segurança e a presença de *deadlocks*.



**Figura 3.12 - Análise das propriedades comportamentais do modelo completo em Rede de Petri do Provedor de Serviços.**

Na Figura 3.12 tem-se a constatação que o modelo em Rede de Petri do Provedor de Serviços trata-se de um modelo limitado, não seguro e que não possui *deadlocks*.

Uma RdP é dita limitada se em cada lugar da rede, o número total de marcas nunca exceder a um inteiro  $k$ . Neste caso a rede é dita  $k$ -limitada. Se a rede for limitada ao inteiro 1, diz-se portanto, que a rede é segura [...],(Bonet et. al., 2007).

A constatação da limitação de uma Rede de Petri indica que os lugares da mesma nunca excedem um número específico de marcações (Cassandras et. al, 2008). No caso do Provedor de Serviços, esse número é 128, ou seja, a constatação da limitação do modelo assegura que a Rede de Petri descrita na Figura 3.8 não permite que existam mais de 128 nós inscritos no leito em um determinado instante.

A segurança de uma Rede de Petri está relacionada à quantidade de marcações presentes em seus lugares. No caso, uma rede segura admite somente a existência de uma marcação em cada lugar da rede. O modelo especificado para o Provedor de Serviços,

portanto, é dito como não seguro, isso em virtude da necessidade de controle do número de inscrições realizadas no leito (que pode chegar a 128), para tanto existe o lugar “Vagas” que possui mais de uma marcação, cujo objetivo é representar à quantidade de inscrições que ainda podem ser efetuadas em um leito. A opção por um modelo não seguro se dá em função do nível de abstração adotado na modelagem do Provedor de Serviços, visto que, vez que o objetivo maior desta especificação foi de modelar um sistema livre de *deadlock*. É importante destacar que um modelo com um nível mais baixo de abstração do Provedor Serviços poderia ter como propriedade uma Rede de Petri segura. Todavia, seria um modelo mais complexo, com baixo de nível de expressão e agregaria pouco ao objeto de estudo dessa dissertação, visto que, para o Provedor de Serviços o mais importante é garantir que este está livre de *deadlock*, fator que representa característica de robustez e disponibilidade - aspectos indispensáveis nos processos de automação hospitalar.

Um *deadlock* caracteriza uma situação de impasse entre dois ou mais processos em que ambos ficam esperando a liberação de recursos atrelados a outros de modo que todos ficam impedidos de continuarem as suas execuções, caracterizando assim, um travamento do sistema (Tanenbaum, 2000). No caso do Provedor de Serviços, o modelo em Rede de Petri apresentado não possui *deadlocks*, ou seja, não existe nenhuma situação em que ocorre a disputa de recursos de modo a gerar uma situação de impasse no sistema como um todo.

A garantia de ausência de *deadlocks* é o ponto mais importante deste modelo, isto porque, essa verificação garante que o sistema não trava em condições normais de execução, o que é uma característica desejada não só em termos de eficiência, como também em virtude da importância que o Provedor de Serviços assume dentro da rede PM-AH.

Dado a verificação dessas propriedades, nota-se que o modelo descrito na Figura 3.8 apresenta características que viabilizam a sua implementação, uma vez que atendem a todas as necessidades impostas pelo PM-AH para o correto funcionamento da rede entre dispositivos biomédicos.

#### 4. ANÁLISE DE DESEMPENHO DO PROVEDOR DE SERVIÇOS

O PS é um dos elementos fundamentais da rede PM-AH, pois é responsável pelo estabelecimento e manutenção da comunicação entre os dispositivos biomédicos (Souza, 2008). Em decorrência da sua importância, é interessante que o Provedor de Serviços apresente um bom desempenho na execução de suas atividades, visto que, são elas as responsáveis pelo cumprimento de uma série de requisitos necessários ao bom funcionamento da rede hospitalar (Valentim et al., 2008).

Esse capítulo aborda a realização de análises de desempenho sobre aspectos relacionados ao PS, como forma de avaliar a eficiência do modelo proposto pelo PM-AH para o Provedor de Serviços.

A primeira análise de desempenho diz respeito ao tempo de processamento gasto por operações de acesso a banco de dados no tratamento de mensagens de requisição de inscrição. Essa análise de desempenho é importante em virtude de boa parte das operações efetuadas durante a inscrição de dispositivos biomédicos estarem relacionadas ao acesso e manipulação de registros armazenados em banco de dados, dessa forma, é importante avaliar o impacto dessas operações na realização de uma inscrição de dispositivo biomédico.

A segunda análise de desempenho diz respeito ao tempo de entrega de mensagens de dados ao Provedor de Serviços. Essa análise de desempenho é importante para avaliação da eficiência da estrutura imposta pelo PM-AH para o recebimento de tais mensagens, visto que, existem enfileiramentos e perda de mensagens na porta do *switch* em que o PS está conectado (Valentim, 2008).

A terceira análise de desempenho diz respeito à avaliação de uma nova proposta para o Provedor de Serviços através do uso de passagem de *token* para coordenar o recebimento de mensagens de dados provenientes dos dispositivos biomédicos. Essa nova proposta tem por objetivo melhorar o desempenho na entrega de mensagens de dados dos dispositivos biomédicos ao Provedor de Serviços utilizando um mecanismo de passagem de mensagens de *token* similar ao que ocorre dentro dos leitos hospitalares durante uma janela síncrona. Dessa forma, foram feitas análises de desempenho para levantar o ganho obtido por essa nova abordagem, em comparação a abordagem proposta pelo PM-AH, e foram levantados quais pontos no PM-AH deveriam ser modificados para que a adoção dessa nova abordagem fosse possível.

#### **4.1. Análise de desempenho do tempo de processamento do acesso aos registros: custo do processamento no banco de dados para efetuar o tratamento de requisições de inscrição**

As mensagens de requisição de inscrição (RMJ) são enviadas pelos dispositivos logo após serem iniciados e tem o objetivo de solicitar ao Provedor de Serviços a permissão para efetuarem a associação em um determinado leito (grupo *multicast*) da rede hospitalar PM-AH (Valentim, 2008).

Ao receber uma RMJ, o Provedor de Serviços deve realizar uma consulta ao banco de dados para saber se o dispositivo solicitante já está inscrito no leito indicado na mensagem de requisição. Caso não esteja, o PS deve realizar mais uma consulta para verificar se o leito desejado já está cheio (Valentim et al., 2008). Se o leito não estiver cheio, o Provedor de Serviços tentará efetuar a inscrição do novo dispositivo, para tanto, deve realizar mais uma consulta para saber se o leito em questão possui um mestre associado. Caso não possua, e o dispositivo solicitante tenha capacidade de processamento suficiente, é feita à sua inscrição na condição de mestre do leito, do contrário, mais uma consulta é necessária para obter um valor de índice de modo que tal dispositivo seja inscrito na condição de escravo (Souza, 2008).

Pelo exposto, é perceptível que na realização de uma inscrição, boa parte do processamento efetuado pelo PS diz respeito à execução de operações de manipulação de informações contidas no banco de dados (o qual foi designado ao armazenamento de registros pertinentes aos estados dos leitos e dispositivos da rede hospitalar) (Valentim et al., 2008). Dessa forma é interessante efetuar uma análise de desempenho do tempo de processamento gasto na realização dessas operações como forma de verificar o impacto dessas no Provedor de Serviços.

Para tanto, foram implementadas rotinas e procedimentos que correspondessem às atividades descritas na Rede de Petri da Figura 3.9 (responsável pelo gerenciamento de inscrição de dispositivos) e foi levantado o tempo de processamento gasto pelas operações dessa Rede que envolviam a manipulação de informações armazenadas em banco.

Vale salientar que nessas implementações utilizou-se a linguagem de programação C++, o banco de dados adotado foi o MySQL *Server* 5.1 e a disposição das tabelas seguiu o modelo descrito na Figura 3.2 (modelo lógico do banco de dados para o Provedor de Serviços).

A seguir têm-se a descrição dos cenários de teste adotados bem como dos resultados obtidos mediante a execução dos mesmos.

#### 4.1.1. Cenários de teste

Para a análise de desempenho foram montados dois cenários de teste compostos por 25 e 50 leitos cada, nos quais, considerou-se a existência de 300 tipos de dispositivos distintos cadastrados no sistema.

A variação do número de leitos é importante, pois permite avaliar o desempenho do tempo de processamento de operações envolvendo o custo dos acessos ao banco, quando esse armazena diferentes volumes de dados, visto que, quanto maior a quantidade de leitos, mais inscrições podem ser realizadas pelo Provedor de Serviços. Neste sentido, este tipo de análise contribuiu para verificar o comportamento do PS, com relação ao custo computacional demandado pelas consultas a base de dados, os quais são provenientes do aumento de dispositivos na rede PM-AH.

Na execução dos cenários de teste foram computados os tempos de processamento das operações envolvendo acesso ao banco de dados para 100.000 solicitações de inscrição. Como forma de facilitar a análise dos resultados obtidos considerou-se que, em todas as solicitações, os dispositivos solicitantes tinham capacidade de processamento suficiente para atuarem como mestre do leito.

Os índices referentes ao tipo do dispositivo e ao leito desejado nas solicitações foram obtidos por meio de funções de geração de números aleatórios, com a adoção de uma metodologia que garantiu a coerência entre os números gerados por essas funções, ou seja, os índices para os tipos de dispositivos deveriam estar entre 0 e 299 (300 dispositivos) e os de índice de leito deveriam estar entre 0 e 24 (25 leitos), no primeiro cenário de testes, e entre 0 e 49 (50 leitos), no segundo cenário de testes.

A máquina utilizada na execução dos cenários de teste foi um Intel Core 2 Duo CPU E7200/2.53 GHz/2 GB RAM/Sistema Operacional Linux (distribuição Ubuntu 9.04) e o levantamento das medidas de desempenho foi realizada com o auxílio do MATLAB R2009A.

#### 4.1.2. Resultados

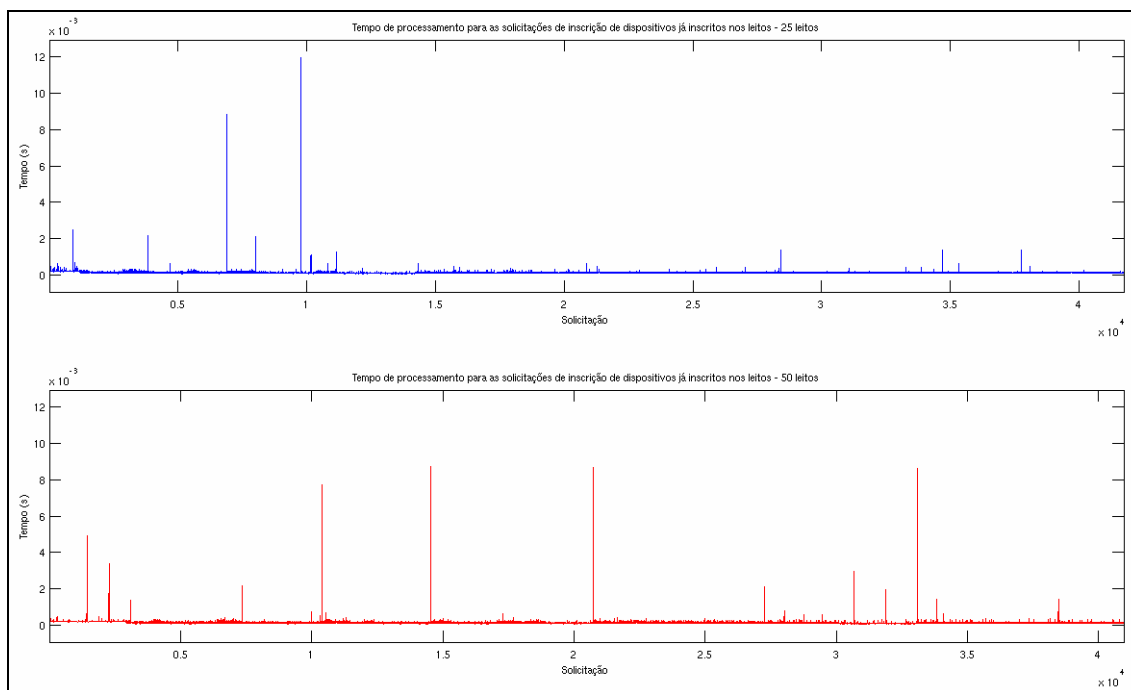
Conforme visto, no tratamento de uma requisição de inscrição, o Provedor de Serviços realiza uma série de operações que envolvem o acesso a registros armazenados em banco de dados. Essa sessão aborda a realização de análises de desempenho sobre os tempos de processamento gastos por essas operações com base na execução dos cenários de teste descritos anteriormente. Sendo assim, as medidas de desempenho levantadas foram:

- **Quantidade de solicitações:** Número de solicitações realizadas;
- **Tempo médio:** Média dos tempos de processamento gastos na execução das

operações envolvendo acesso ao banco de dados;

- **Tempo máximo:** Maior tempo de processamento gasto na execução das operações envolvendo acesso ao banco de dados;
- **Tempo mínimo:** Menor tempo de processamento gasto na execução das operações envolvendo acesso ao banco de dados;

A primeira análise de desempenho aferiu os tempos de processamento gastos e as medidas de desempenho relacionadas às requisições de inscrição, dentre as 100.000 solicitações realizadas, que falharam em virtude do dispositivo solicitante já estar associado ao leito desejado. A descrição dos tempos de processamento pode ser vista na Figura 4.1, enquanto que os valores das medidas de desempenho levantadas na execução dos dois cenários de teste podem ser vistos na Tabela 4.1.



**Figura 4.1 – Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falharam em virtude dos dispositivos já estarem inscritos nos leitos.**

**Tabela 4.1 - Medidas de desempenho sobre os tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falharam em virtude dos dispositivos já estarem inscritos nos leitos.**

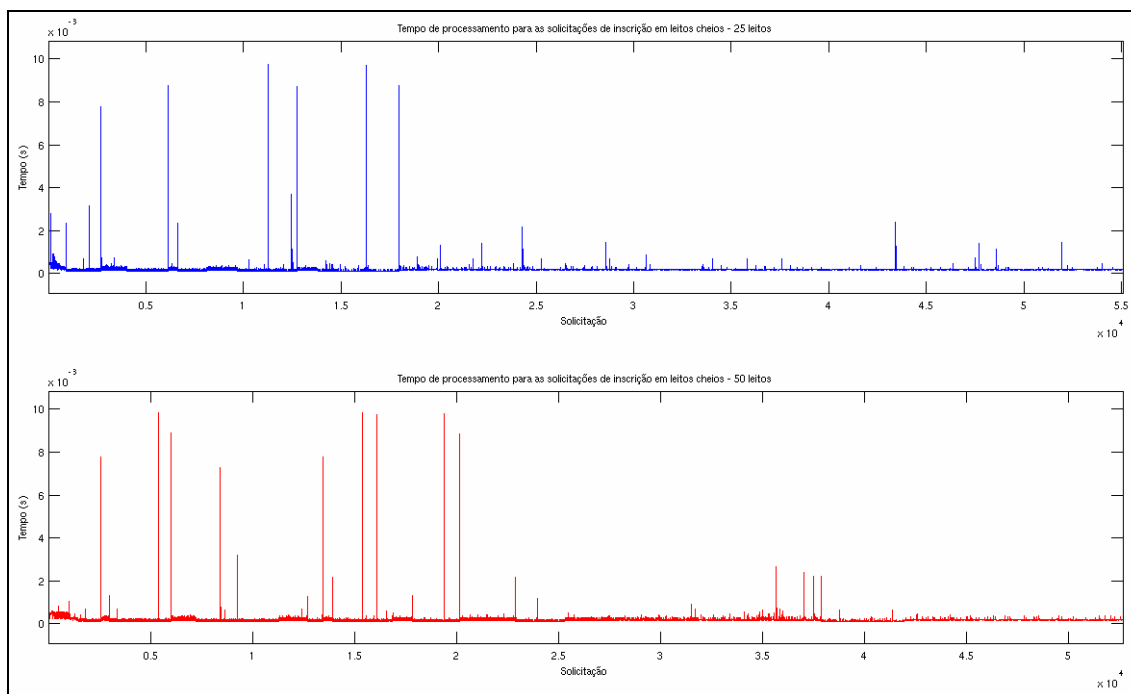
<b>Medidas de Desempenho</b>	<b>25 leitos</b>	<b>50 leitos</b>
Quantidade de solicitações	41.733 solicitações	40.954 solicitações
Tempo médio	$1.1892 \times 10^{-4}$ s	$1.2389 \times 10^{-4}$ s
Tempo máximo	0.0119 s	0.0087 s
Tempo mínimo	$6.4000 \times 10^{-5}$ s	$6.4000 \times 10^{-5}$ s

Pelos resultados acima é possível observar que, quando uma solicitação de inscrição falha em virtude de o dispositivo solicitante estar inscrito no leito desejado, o tempo de processamento gasto pelas operações que envolvem acesso ao banco de dados é bem pequeno. Esse baixo tempo pode ser justificado, pois, nessa situação, apenas dois acessos ao banco são realizados: um para verificar se o dispositivo e o leito estão cadastrados no sistema e outro para verificar se o dispositivo está inscrito no leito. Desse modo, em virtude do baixo tempo consumido por acessos simples ao banco, o tempo de processamento gasto torna-se pequeno em decorrência do número de acessos realizados.

Outro aspecto importante denotado na análise em relação às medidas de desempenho apresentada na Tabela 4.1 é a similaridade dos valores obtidos para os tempos médio, máximo e mínimo na execução dos dois cenários de teste. A presença desse aspecto é importante, pois assegura que, para as solicitações de inscrição que falham em virtude dos dispositivos já estarem inscritos nos leitos, o comportamento do Provedor de Serviços será o mesmo independentemente da quantidade de informações armazenadas em banco.

A segunda análise de desempenho aferiu os tempos de processamento e as medidas de desempenho relacionadas às requisições de inscrição, dentre as 100.000 solicitações realizadas, que falharam em virtude do leito no qual o biodispositivo estava tentando fazer a inscrição estar cheio. A descrição dos tempos de processamento é ilustrada na Figura 4.2, enquanto que os valores das medidas de desempenho obtidas na execução dos dois cenários de teste podem ser vistos na Tabela 4.2.





**Figura 4.2 - Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falham em virtude dos leitos desejados estarem cheios.**

**Tabela 4.2 - Medidas de desempenho sobre os tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que falham em virtude dos leitos desejados estarem cheios.**

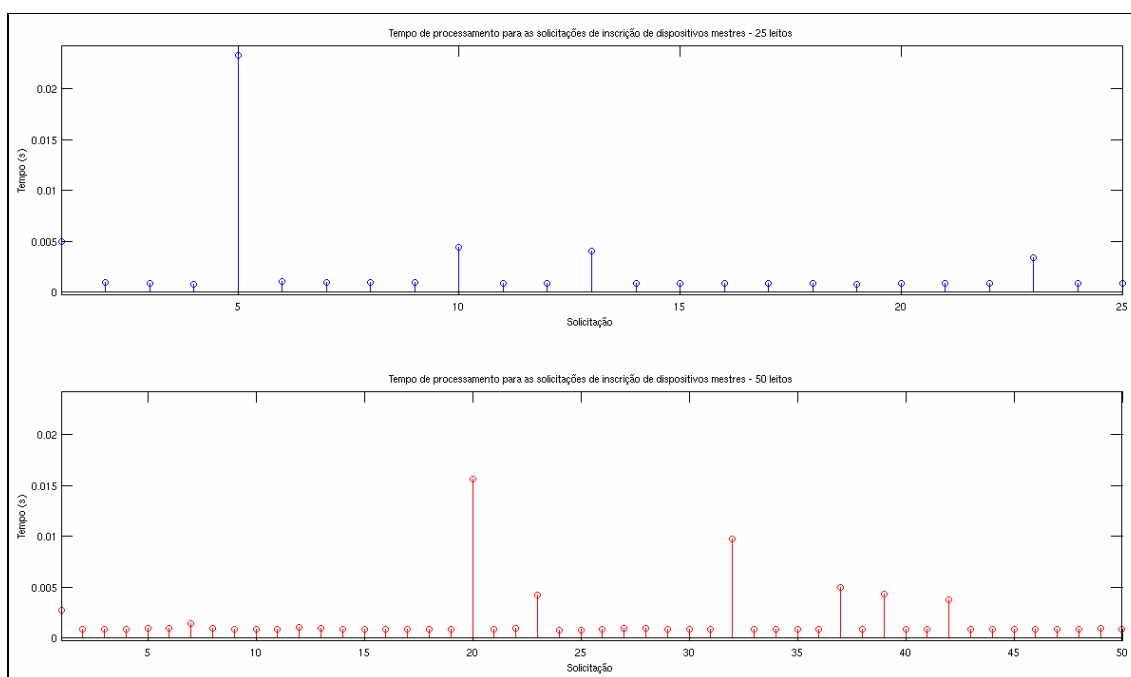
<b>Medidas de Desempenho</b>	<b>25 leitos</b>	<b>50 leitos</b>
Quantidade de solicitações	55.067 solicitações	52.646 solicitações
Tempo médio	$1.7413 \times 10^{-4}$ s	$1.7520 \times 10^{-4}$ s
Tempo máximo	0.0098 s	0.0098 s
Tempo mínimo	$9.9000 \times 10^{-5}$ s	$9.9000 \times 10^{-5}$ s

Observando os resultados da Figura 4.2 é perceptível que o comportamento do Provedor de Serviços manteve-se constante, mesmo com a variação na quantidade de informações armazenadas no banco. Além da manutenção de comportamento, os resultados apresentam outro aspecto interessante: um aumento nos valores dos tempos médio, máximo em comparação aos medidos na análise de desempenho anterior.

A justificativa para esse aumento de tempo está no fato do processamento das solicitações de inscrição em leitos cheios exigir uma consulta a mais do que as solicitações de inscrição de dispositivos que já estão associados ao leito desejado. Isso ocorre, porque, além das consultas para verificar a validade do leito e do dispositivo indicados na RMJ, e para

saber se o dispositivo solicitante já está inscrito no leito, é realizada também uma consulta a mais para identificar se o leito em questão está cheio.

A terceira análise de desempenho levantou os tempos de processamento gastos e as medidas de desempenho relacionadas às requisições, dentre as 100.000 solicitações realizadas, que resultaram na inscrição de um dispositivo na condição de mestre do leito. A descrição dos tempos de processamento pode ser vista na Figura 4.3, enquanto que os valores das medidas de desempenho levantadas na execução dos dois cenários de teste podem ser vistos na Tabela 4.3.



**Figura 4.3 - Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de mestre do leito.**

**Tabela 4.3 - Medidas de desempenho referentes aos tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de mestre do leito.**

Medida de desempenho	25 leitos	50 leitos
Quantidade de solicitações	25 solicitações	50 solicitações
Tempo médio	0.0023 s	0.0017 s
Tempo máximo	0.0233 s	0.0156 s
Tempo mínimo	$7.6900 \times 10^{-4}$ s	$7.7000 \times 10^{-4}$ s

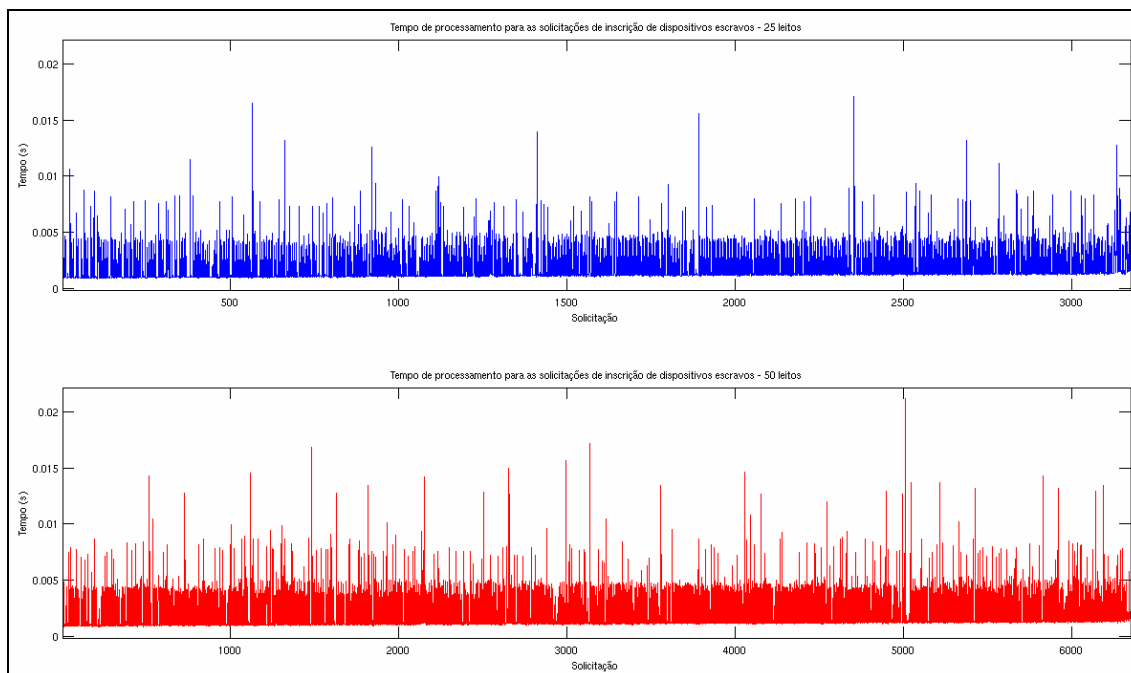
Com base nos resultados descritos na Tabela 4.3 é possível notar que o comportamento do Provedor de Serviços na realização das operações de acesso ao banco de dados, manteve-se aproximadamente constante, tanto na a execução do cenário de teste

composto por 25 leitos, como na execução do cenário de teste composto por 50 leitos. Sendo assim, pode-se afirmar que o desempenho do PS na realização dessas operações não foi influenciado pelo volume de dados armazenados no banco.

Outro aspecto que pode ser observado é que, as medidas de desempenho descritas na Tabela 4.3 são maiores em comparação aos valores presentes na Tabela 4.2 e na Tabela 4.1. Esse crescimento se dá em virtude do aumento de consultas efetuadas ao banco de dados quando ocorre a inscrição de um dispositivo na condição de mestre do leito, visto que, todas as consultas realizadas quando ocorre falha de inscrição (seja pelo dispositivo já estar inscrito ou pelo leito estar cheio) são realizadas novamente e somado a essas é realizada mais uma consulta destinada a identificar se o leito em questão já possui um mestre associado.

Neste contexto, somado ao tempo adicional dessa nova consulta, existe também outro fator que é o tempo de inserção do registro da inscrição do dispositivo na tabela de associações do banco. Sendo assim, o tempo gasto nessa operação também contribui para o aumento dos valores presentes na Tabela 4.3.

A quarta análise de desempenho levantou os tempos de processamento e as medidas de desempenho relacionadas às requisições, dentre as 100.000 solicitações realizadas, que resultaram na inscrição de um dispositivo biomédico na condição de escravo do leito. A descrição dos tempos de processamento pode ser vista na Figura 4.4, enquanto que os valores das medidas de desempenho aferidas na execução dos dois cenários de teste podem ser vistos na Tabela 4.4.



**Figura 4.4 - Tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de escravo do leito.**

**Tabela 4.4 - Medidas de desempenho referentes aos tempos de processamento gastos por operações que efetuam o acesso ao banco de dados nas solicitações de inscrição que resultam nas associações de um dispositivo na condição de mestre do leito.**

<b>Medida de desempenho</b>	<b>25 leitos</b>	<b>50 leitos</b>
Quantidade de solicitações	3.175 solicitações	6.350 solicitações
Tempo médio	0.0020 s	0.0020 s
Tempo máximo	0.0171 s	0.0212 s
Tempo mínimo	$8.78 \times 10^{-4}$ s	$8.68 \times 10^{-4}$ s

Os resultados descritos na Tabela 4.4 denotam que, assim como nas situações anteriores, as medidas de desempenho foram muito similares nos dois cenários de teste considerados. Esse resultado é relevante, pois, demonstra que, o Provedor de Serviços mantém um comportamento constante e estável na execução das operações que demandam acesso a registros do banco de dados, cujo objetivo foi efetuar a inscrição de novos dispositivos.

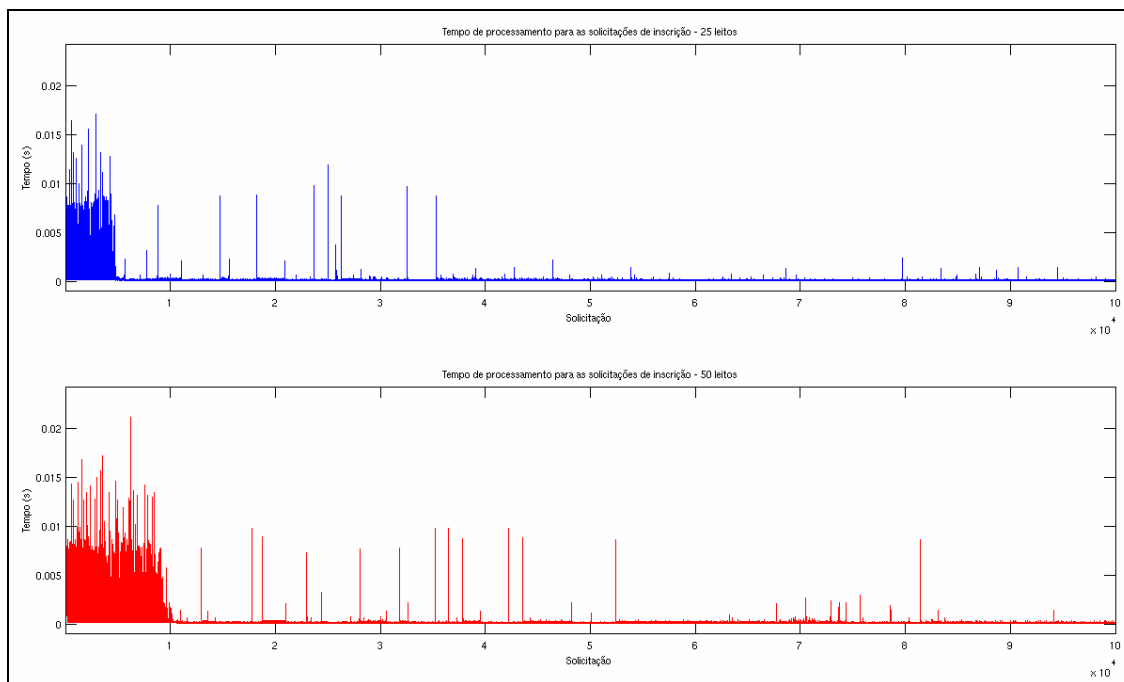
Outro fato que pode ser observado também na Tabela 4.4 é o aumento nos tempos medidos em relação aos valores levantados para as situações anteriores (falha de inscrição pelo fato do dispositivo já estar inscrito, falha de inscrição pelo fato do leito estar cheio ou inscrição de dispositivos na condição de mestre dos leitos). Esse aumento é justificado pela

necessidade de consultas adicionais quando comparado às situações anteriores, visto que é necessário realizar consultas para identificar se o dispositivo pode ser inscrito como escravo (uma vez que, como forma de garantir o correto funcionamento da rede hospitalar, o ideal é que apenas 127 dispositivos podem ser inscritos na condição de escravo em cada leito) e para obter um índice válido, pelo qual o dispositivo em questão será diferenciado no leito.

Além disso, a inscrição de um dispositivo escravo demanda um custo computacional para inserir um registro na tabela de associações referente à inscrição do dispositivo solicitante. Sendo assim, esse tempo também influenciou no aumento de valores computados nas medidas de desempenho levantadas.

A próxima análise de desempenho apresenta os tempos de processamento gasto nas 100.000 solicitações de inscrição segundo a ordem em que essas foram realizadas. Sendo assim, os resultados descritos na Figura 4.5 e na Tabela 4.5 abordam, respectivamente, os tempos de processamento na realização de todas as solicitações de inscrição e as medidas de desempenho levantadas com base nos tempos dessas solicitações.

A análise do gráfico da Figura 4.5 remete pontos interessantes que devem ser avaliados. Pela análise do gráfico é possível perceber que, no início, os tempos de processamento por operações que realizam acesso ao banco possuem valores altos e a partir de certo ponto tais valores diminuem consideravelmente. A presença desse comportamento é justificada pelo fato dos leitos estarem inicialmente vazios, sendo assim, a probabilidade que uma solicitação de inscrição falhe no início da execução dos cenários de teste é muito pequena em decorrência do fato que os aspectos de geração de falha (dispositivo inscrito ou leito cheio) não estão presentes ainda. Sendo assim, o tempo de processamento gasto na etapa inicial equivale ao tempo de inscrição de dispositivos mestres e escravos, que conforme verificado, são os de maior custo computacional.



**Figura 4.5 - Tempos de processamento gasto na realização das operações de acesso ao banco de dados para todas as solicitações de inscrição realizadas.**

**Tabela 4.5 - Medidas de desempenho dos tempos de processamento gasto na realização das operações de acesso ao banco de dados para todas as solicitações de inscrição realizadas.**

<b>Medida de desempenho</b>	<b>25 leitos</b>	<b>50 leitos</b>
Quantidade de solicitações	100.000 solicitações	100.000 solicitações
Tempo médio	$2.1039 \times 10^{-4}$ s	$2.7043 \times 10^{-4}$ s
Tempo máximo	0.0233 s	0.0212 s
Tempo mínimo	$6.4000 \times 10^{-5}$ s	$6.4000 \times 10^{-5}$ s

À medida que as inscrições vão acontecendo, as tabelas do banco vão sendo preenchidas, e os leitos vão ficando cheios, fator que aumenta a probabilidade de uma solicitação de inscrição falhar em decorrência do dispositivo solicitante já estar inscrito ou do leito estar cheio. Desse modo o tempo de processamento na etapa final da execução dos cenários de teste passa a ser o tempo necessário para processar solicitações de inscrição de dispositivos já inscritos no leito ou em leitos cheios os quais, conforme visto, tem um menor custo computacional.

Outro aspecto interessante do gráfico da Figura 4.5 é que o intervalo de solicitações em que o tempo de processamento é mais elevado, decorrente da inscrição de dispositivos mestre e escravos, é maior no cenário de teste composto por 50 leitos do que no cenário de

teste composto por 25 leitos. Esse aspecto é justificado pelo fato do cenário de testes composto por 50 leitos permitir a realização de um número maior de inscrições em comparação ao cenário de testes composto por 25 leitos. Como a fase em que as inscrições ocorrem é a etapa inicial da execução dos cenários de teste, então o intervalo de tempo correspondente a essa etapa torna-se maior no cenário de teste composto por 50 leitos em decorrência do número maior de inscrições que podem ser realizadas.

## **4.2. Análise de desempenho do Provedor de Serviços na rede PM-AH: transferência de mensagens de dados**

Dentre as tarefas realizadas pelo Provedor de Serviços uma das mais importantes é o cadastro de dados referentes às medidas dos sinais vitais dos pacientes. A boa execução dessa atividade é importante, pois demanda aplicações de predição de diagnóstico, levantamento de histórico dentre outras (Valentim, 2008).

Em virtude de toda mensagem que trafega na rede ser replicada na porta do *switch* na qual o Provedor de Serviços está conectado, é possível que ocorram enfileiramentos e percas de pacotes nessa porta (Valentim, 2008). Sendo assim, é interessante avaliar o desempenho dessa transmissão, de modo a gerar sobrecarga de processamento e de dados de comunicação no Provedor de Serviços.

Nesse contexto, essa sessão realiza uma análise comportamental do Provedor de Serviços no recebimento de mensagens de dados. As análises foram feitas por meio de simulações executadas na ferramenta *Network Simulator 2* (NS 2) versão 2.30 que permite a simulação de uma grande quantidade de tecnologias de rede (Coutinho, 2003).

A seguir são realizadas as descrições dos cenários de testes utilizados e dos resultados obtidos nas simulações. Ao apresentar os resultados é feita uma análise dos impactos que a estrutura proposta pelo PM-AH para o Provedor de Serviços pode ocasionar na qualidade da transmissão das mensagens de dados.

### **4.2.1. Cenários de teste**

Para a análise de desempenho foram considerados quatro cenários de teste cada um com um número de leitos diferente. A variação no número de leitos tem como objetivo proporcionar uma análise de desempenho quando a rede é submetida a diferentes níveis de tráfego, uma vez que, aumentando o número de leitos aumenta-se o número de dispositivos, fator que ocasiona um fluxo de mensagens mais intenso (Souza, 2008).

Dito isto, foram projetadas três simulações considerando 10, 25, 50 e 60 leitos. Os parâmetros utilizados nessas simulações foram:

- **Número de dispositivos por leito:** 6 dispositivos;
- **Quantidade de ciclos temporais adotados:** 1000 ciclos;
- **Taxa de transmissão da rede:** 100 Mbps (*Fast-Ethernet*);
- **Tamanho do buffer de mensagens nas portas do switch:** 50 mensagens;
- **Probabilidade de ocorrência de alarmes:** Distribuição uniforme com probabilidade de 1% para a ocorrência de alarmes nos nós do leito dentro de um ciclo temporal (fator baseado nos experimentos realizados por Gullikson (Gullikson, 2005));

Além das especificações, outro ponto interessante nas simulações executadas foi o uso de fatores representativos dos tempos de subida e descida na pilha de protocolos. Os valores adotados para tais tempos foram baseados no trabalho de Viégas e Valentim (Viégas e Valentim, 2006) no qual eles verificaram uma variação entre 0,71 e 1,57 microssegundos na transferência de pacotes de 64 *bytes*.

A adoção desses valores é interessante, pois traz aspectos verificados em ambientes experimentais ao trabalho melhorando a precisão dos modelos desenvolvidos no simulador (Jain, 1991).

Todas as simulações no *Network Simulator* foram projetadas para que gerassem um arquivo de *log* chamado *trace file*. Esse arquivo contém a descrição de todos os eventos que ocorrem na rede ao longo da simulação (Coutinho, 2003).

Como esse arquivo pode chegar a vários *Megabytes* de tamanho (Coutinho, 2003) o processamento deles foi realizado por rotinas desenvolvidas em JAVA e as métricas referentes à qualidade da transmissão foram levantadas com o auxílio do MATLAB R2006b.

#### **4.2.2. Resultados**

Os resultados mostrados a seguir foram obtidos a partir da simulação dos cenários de teste descritos anteriormente. Esses resultados correspondem a métricas de desempenho da transmissão de mensagens de dados dos dispositivos biomédicos para o Provedor de Serviços.

A obtenção dessas métricas foi feita a partir do monitoramento de uma transmissão específica. Essa transmissão foi das mensagens de dados que partiram do dispositivo de índice 3 do leito 1 para o Provedor de Serviços.

Os parâmetros de desempenho levantados nas simulações foram:

- **Atraso médio:** Média dos tempos decorridos entre o envio de mensagens



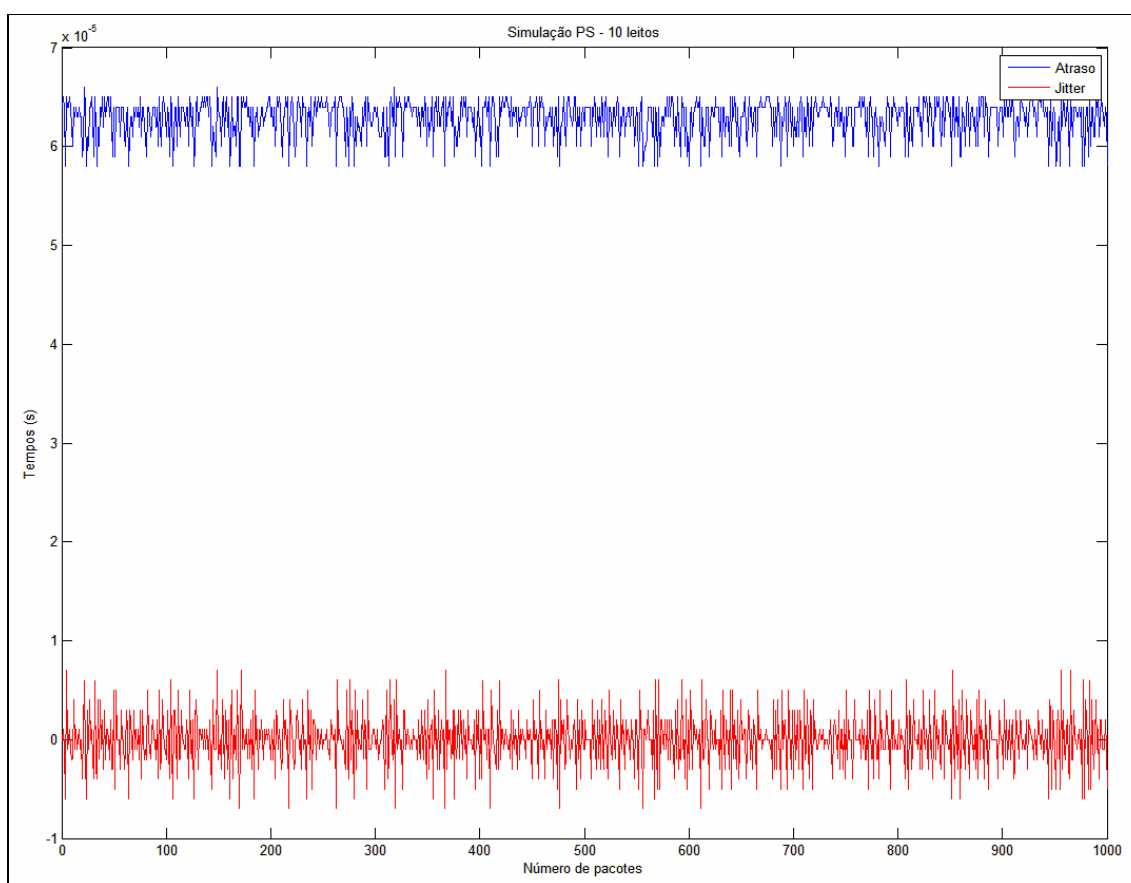
de dados do dispositivo biomédico monitorado e a chegada dessas no Provedor de Serviços.

- **Jitter médio:** Média das variações do atraso de todas as transmissões monitoradas.
- **Desvio padrão:** Dispersão estatística dos tempos de atraso na transmissão de mensagens de dados.
- **Número de pacotes perdidos na transmissão:** Quantidade de mensagens de dados perdidas na transmissão monitorada.
- **Número de pacotes perdidos na rede:** Quantidade de mensagens perdidas em todas as transmissões efetuadas.
- **Número de alarmes gerados:** Quantidade de mensagens de alarme geradas por todos os dispositivos da rede.
- **Número de pacotes gerados:** Quantidade de mensagens enviadas ao Provedor de Serviços na simulação (soma das quantidades de mensagens de *token*, dados e alarmes).

A primeira simulação executada foi a do cenário de teste que continha 10 leitos de 6 dispositivos cada (60 dispositivos no total). As medidas de desempenho obtidas nessa simulação estão descritas na Tabela 4.6 enquanto que o gráfico da Figura 4.6 ilustra o atraso e o *jitter* do envio de cada pacote de dados separado.

**Tabela 4.6 - Parâmetros de desempenho da transmissão de dados (simulação para 10 leitos de 6 dispositivos cada).**

Parâmetros de desempenho	Valor
Atraso médio	$6.2918 \times 10^{-5} \text{s}$
<i>Jitter</i> Médio	$6.0000 \times 10^{-9} \text{s}$
Desvio padrão	$1.9102 \times 10^{-6} \text{s}$
Número de pacotes perdidos (transmissão)	0 pacotes
Número de pacote perdidos (rede)	0 pacotes
Número de alarmes gerados	683 pacotes
Número de pacotes gerados	120.683 pacotes



**Figura 4.6 - Atraso e *Jitter* da transmissão de dados (simulação para 10 leitos de 6 dispositivos cada).**

A análise dos resultados apresentados pela simulação remete pontos interessantes da transmissão de mensagens de dados entre os dispositivos biomédicos e o Provedor de Serviços.

O primeiro ponto diz respeito ao fato dessa transmissão não ser determinística, isso pode ser verificado pela alta variação do *jitter* no gráfico da Figura 4.6. O não-determinismo nesse caso é justificado pelo fato do Provedor de Serviços pertencer a todos os grupos

*multicast* da rede hospitalar e com isso receber toda mensagem que trafega pela mesma (Valentim, 2008). Como os leitos operam em paralelo, no instante em que um nó está enviando sua mensagem de dados, outros nós pertencentes a outros leitos também o fazem, com isso, ao redirecionar as mensagens ao Provedor de Serviços ocorrerão enfileiramentos na porta do *switch* em que esse está conectado.

O determinismo na transmissão de dados é um fator importante no PM-AH pelo fato dos dispositivos biomédicos poderem realizar ações de controle mediante o recebimento de mensagens de dados, sendo assim, essas mensagens devem chegar nos dispositivos biomédicos em instantes de tempo bem definidos (Valentim et. al., 2008).

Embora o determinismo seja um fator importante, a ausência do mesmo terá grande impacto apenas dentro do leito, na medida que, apenas os dispositivos biomédicos realizam ações de controle baseadas em mensagens de dados, enquanto que, o Provedor de Serviços simplesmente realiza o cadastro das informações presentes em tais mensagens (Valentim et. al., 2008), desse modo, o não-determinismo observado na Figura 4.6 não interfere nas sub-redes geradas pelo PM-AH, ou seja, as redes de monitoramento de pacientes não são afetadas.

Dessa forma, o não-determinismo na entrega de mensagens de dados ao PS não tem grande impacto no desempenho da rede, pois a principal preocupação do Provedor de Serviços com relação a tais mensagens é que estas sejam recebidas e que possam ser registradas as informações contidas nelas.

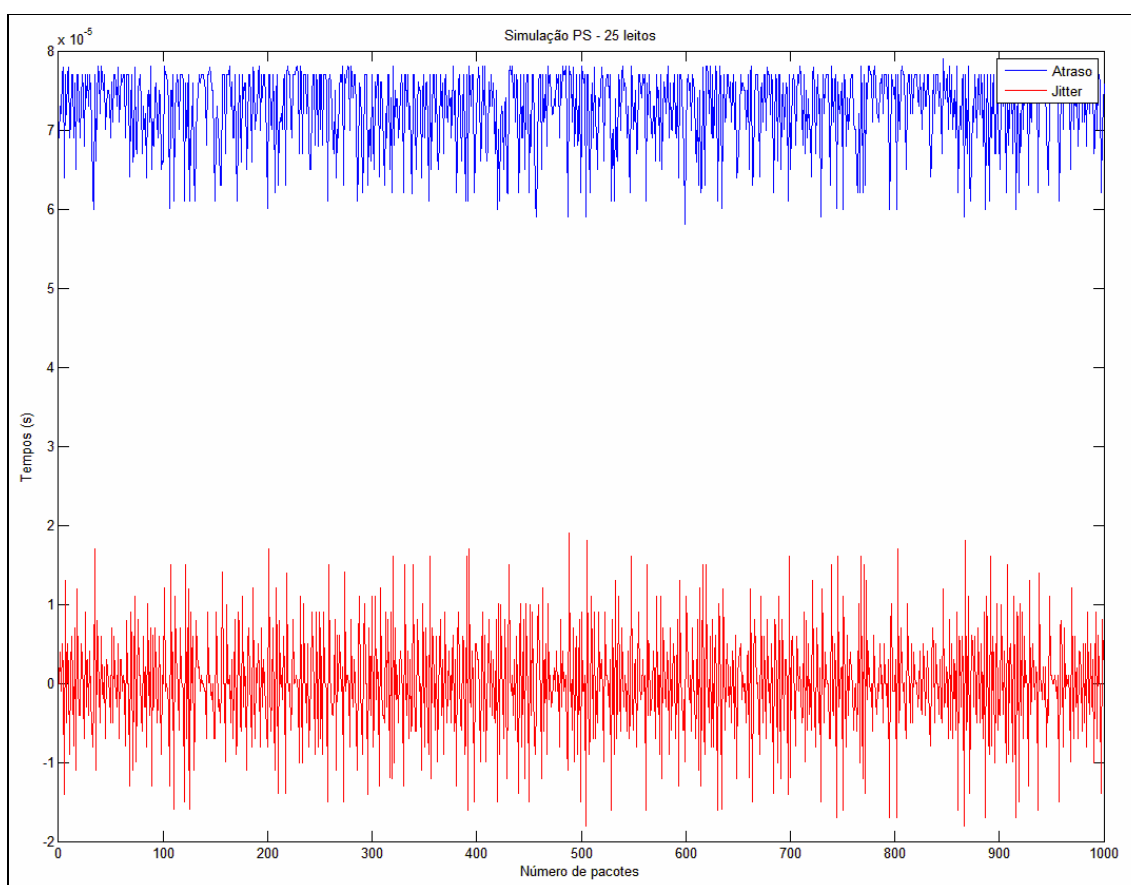
Nesse ponto, entra o segundo aspecto verificado na simulação: a ausência de perda de mensagens de dados. Esse aspecto é importante pelo fato de assegurar que toda mensagem enviada foi entregue ao Provedor de Serviços, de modo que, todos os dados medidos pelos dispositivos biomédicos foram cadastrados. Dessa forma, os objetivos na comunicação de dados entre os nós da rede e o PS foram alcançados.

A próxima simulação executada é a do cenário de teste contendo 25 leitos com 6 dispositivos cada (150 dispositivos ao todo). Essa simulação é interessante pelo fato de permitir uma avaliação da transmissão de mensagens de dados quando o número de dispositivos na rede aumenta.

As medidas de desempenho obtidas nessa simulação são descritas na Tabela 4.7 enquanto que o gráfico da Figura 4.7 ilustra o atraso e o *jitter* do envio de cada pacote de dados na transmissão monitorada (dispositivo de índice 3 do leito 1 para o Provedor de Serviços).

**Tabela 4.7 - Parâmetros de desempenho da transmissão de dados (simulação para 25 leitos com 6 dispositivos cada).**

Parâmetros de desempenho	Valor
Atraso médio	$7.2671 \times 10^{-5} \text{s}$
<i>Jitter</i> Médio	$7.0000 \times 10^{-9} \text{s}$
Desvio padrão	$4.9664 \times 10^{-6} \text{s}$
Número de pacotes perdidos (transmissão)	0 pacotes
Número de pacote perdidos (rede)	0 pacotes
Número de alarmes gerados	1.655 pacotes
Número de pacotes gerados	301.655 pacotes



**Figura 4.7 - Atraso e *Jitter* da transmissão de dados (simulação para 25 leitos com 6 dispositivos cada).**

Nessa simulação, assim como na anterior, foi verificado que as transmissões de dados são não-determinísticas e que não houve perda de mensagens. Além desses aspectos, um ponto dessa simulação que merece destaque é o fato dela ter apresentado um aumento no atraso médio da transmissão quando comparado ao caso em que eram considerados apenas 10 leitos com 6 dispositivos cada (atraso passou de  $62.9 \mu\text{s}$  para  $72,7 \mu\text{s}$ ). Esse aumento no atraso decorre da maior quantidade de dispositivos na rede operando em paralelo o que implica

numa maior quantidade de mensagens sendo enviadas ao Provedor de Serviços no mesmo instante de tempo. Com isso, têm-se maiores enfileiramentos na porta do *switch* a qual o PS está conectado.

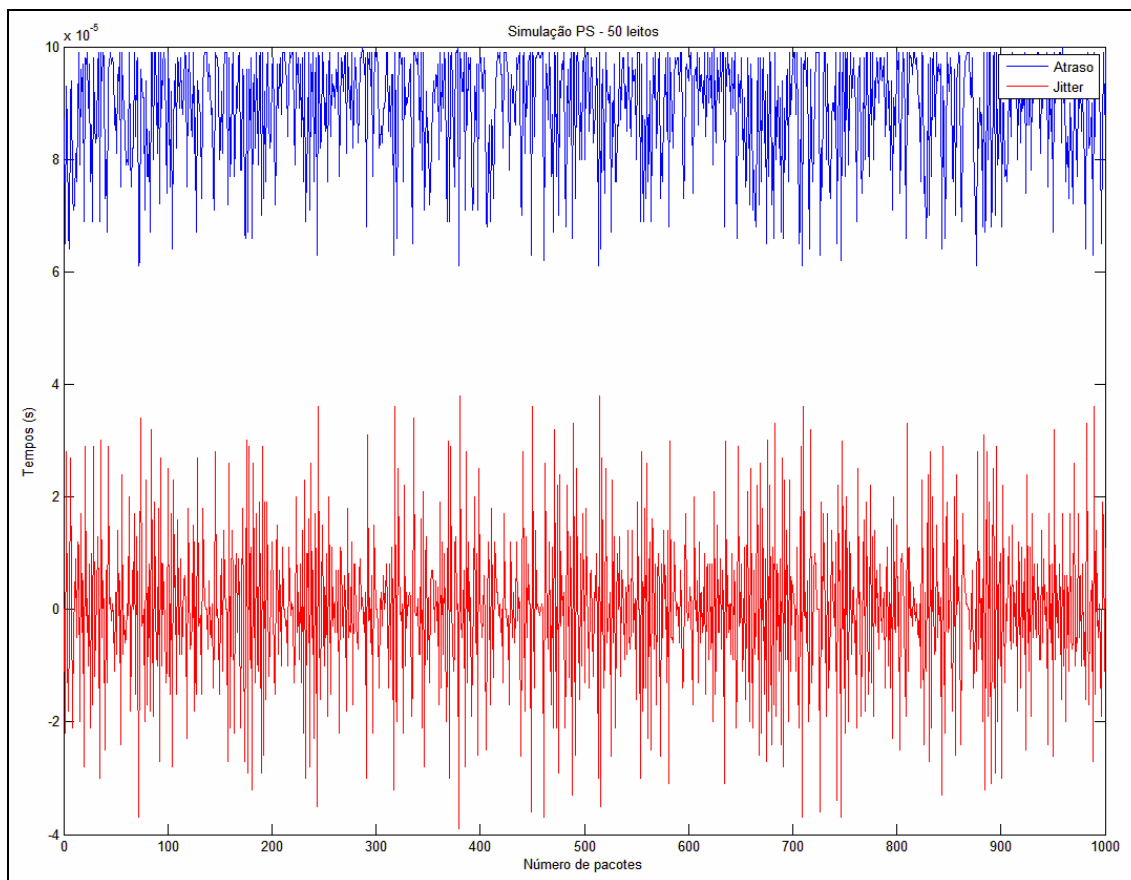
Vale salientar que esse aumento no atraso médio não tem grande impacto na transmissão de mensagens de dados ao Provedor de Serviços na medida em que não houve perda de pacotes, pois garante que o PS cadastrou todos os dados referentes às medições efetuadas pelos dispositivos biomédicos da rede.

A próxima simulação corresponde ao cenário de teste que contém 50 leitos com 6 dispositivos cada (300 dispositivos ao todo). Essa simulação é interessante pelo fato de permitir uma avaliação de desempenho na qual a rede é submetida ao número máximo de mensagens suportado pelo *buffer* das portas do *switch*. Esse aspecto é justificado pelo fato da simulação ter considerado que o *buffer* presente na porta do *switch* poderia armazenar no máximo 50 mensagens, sendo assim, como os 50 leitos do cenário de teste rodam em paralelo, em um dado instante de tempo um dispositivo de cada leito pode enviar sua mensagem de dados totalizando assim 50 mensagens a serem enviadas ao PS (cenário do pior caso), com isso, a fila de *buffer* atinge a sua capacidade máxima.

As medidas de desempenho obtidas na simulação são descritas na Tabela 4.8 enquanto que o gráfico da Figura 4.8 ilustra o atraso e o *jitter* do envio de cada pacote de dados na transmissão monitorada (dispositivo de índice 3 do leito 1 para o Provedor de Serviços).

**Tabela 4.8 - Parâmetros de desempenho da transmissão de dados (simulação para 50 leitos com 6 dispositivos cada).**

<b>Parâmetros de desempenho</b>	<b>Valor (PS sem passagem de <i>token</i>)</b>
Atraso médio	$8.893 \times 10^{-5} \text{ s}$
<i>Jitter</i> Médio	$1.2000 \times 10^{-8} \text{ s}$
Desvio padrão	$1.0183 \times 10^{-5} \text{ s}$
Número de pacotes perdidos (transmissão)	0 pacotes
Número de pacotes perdidos (rede)	0 pacotes
Número de alarmes gerados	3.341 pacotes
Número de pacotes gerados	603.341 pacotes



**Figura 4.8 - Atraso e Jitter da transmissão de dados (simulação para 50 leitos com 6 dispositivos cada).**

Nessa simulação, assim como na anterior, aspectos como aumento no atraso médio da transmissão (de  $72.7\mu\text{s}$  na simulação com 25 leitos para  $88.9\mu\text{s}$  na simulação com 50 leitos) e não-determinismo na entrega das mensagens de dados foram verificados.

Além desses aspectos, outro que também foi verificado foi a inexistência de perda de mensagens de dados durante a simulação. Nesse ponto, é possível afirmar que quando a rede PM-AH opera com a carga máxima suportada pelas portas do *switch* utilizado não ocorre perda de mensagens de dados, com isso o Provedor de Serviços é capaz de cadastrar todas as medições efetuadas pelos dispositivos biomédicos durante o processo de comunicação.

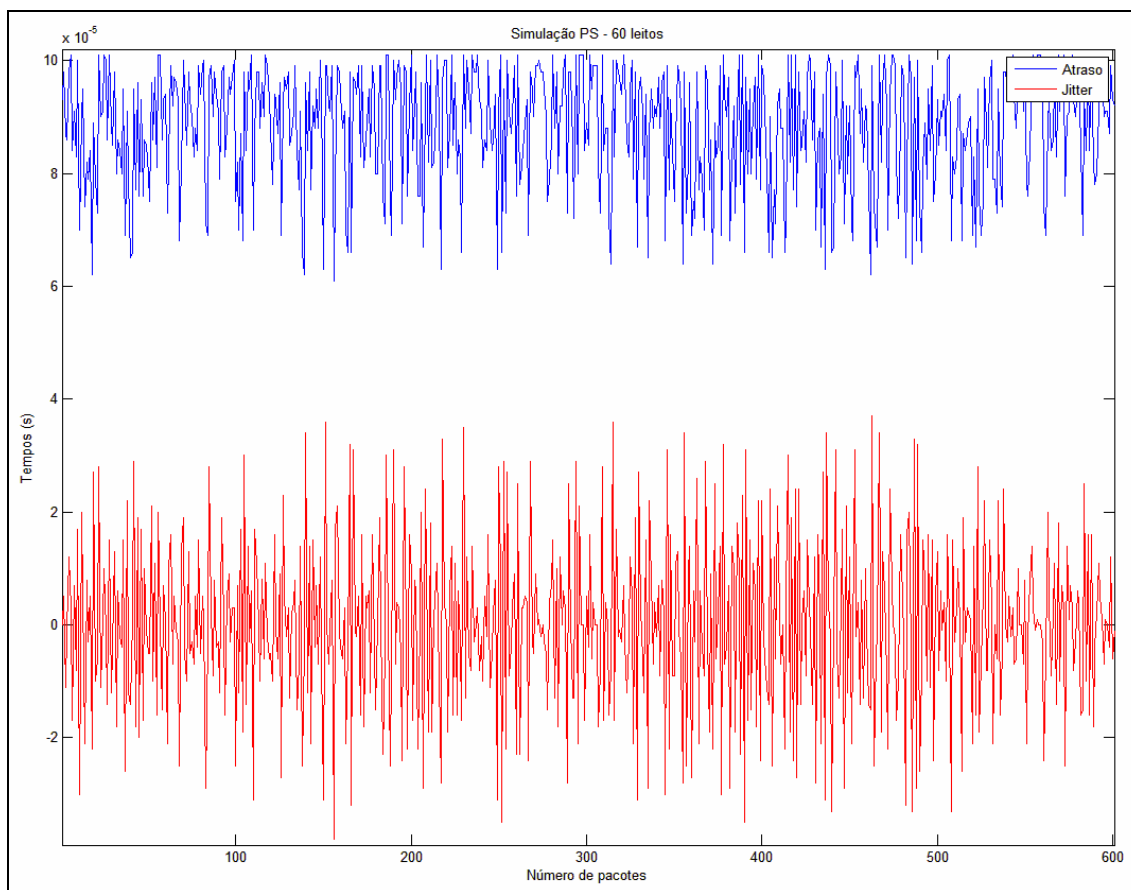
Nesse aspecto, uma análise interessante é a do cenário de teste que contém 60 leitos com 6 dispositivos cada (360 dispositivos no total). Nessa análise é possível perceber que o fluxo de mensagens de dados gerados em um instante de tempo ultrapassa a capacidade de armazenamento presente no *buffer* da porta do *switch* na qual o Provedor de Serviços está conectado. A justificativa para esse fato é a mesma para o cenário de 50 leitos, ou seja, como os leitos operam em paralelo, em um dado instante de tempo é possível que um dispositivo de

cada leito possa enviar uma mensagem de dados totalizando assim 60 mensagens a serem entregues ao Provedor de Serviços (este cenário é o pior caso). Como o *buffer* adotado na simulação tem o tamanho de 50 mensagens, ocorrerão perdas de mensagens em decorrência do descarte efetuado pelo *switch*. As mensagens descartadas serão as que chegam quando o *buffer* de mensagens da porta em que o PS está conectado está cheio. Mesmo, sabendo previamente da ocorrência da perda de mensagens neste cenário que é o de pior caso, foi interessante realizar a simulação com vistas a observar o comportamento temporal do protocolo no que diz respeito ao PS.

Sendo assim, as medidas de desempenho obtidas na simulação são descritas na Tabela 4.9 enquanto que o gráfico da Figura 4.9 ilustra o atraso e o *jitter* do envio de cada pacote de dados na transmissão monitorada (dispositivo de índice 3 do leito 1 para o Provedor de Serviços).

**Tabela 4.9 - Parâmetros de desempenho da transmissão de dados (simulação para 60 leitos com 6 dispositivos cada).**

<b>Parâmetros de desempenho</b>	<b>Valor (PS sem passagem de <i>token</i>)</b>
Atraso médio	$8.8115 \times 10^{-5} \text{s}$
<i>Jitter</i> Médio	$1.6639 \times 10^{-9} \text{s}$
Desvio padrão	$1.0769 \times 10^{-5} \text{s}$
Número de pacotes perdidos (transmissão)	399 pacotes
Número de pacote perdidos (rede)	53.983 pacotes
Número de alarmes gerados	3.984 pacotes
Número de pacotes gerados	723.984 pacotes



**Figura 4.9 - Atraso e *Jitter* da transmissão de dados (simulação para 60 leitos com 6 dispositivos cada).**

Analisando os resultados da simulação verifica-se que houve uma perda de 399 nas mensagens enviadas na transmissão monitorada (dispositivo de índice 3 do leito 1 para o Provedor de Serviços) e de 53983 mensagens de dados entre todas as transmissões efetuadas na rede.

A perda de mensagens de dados é um fator indesejável, na medida em que o PS passa a não realizar o cadastro de todas as informações pertinentes as medições dos sinais vitais dos pacientes monitorados.

Embora seja um fator indesejável, a perda de mensagens pode não gerar um impacto tão negativo no desempenho da rede PM-AH, visto que, os sinais vitais monitorados pelos dispositivos biomédicos evoluem significativamente em grandes intervalos de tempo (tem um processo lento, quando comparado com as redes de controle da automação industrial), enquanto que os ciclos temporais são rápidos (períodos de amostragem), ou seja, existe muita redundância promovida pela rede PM-AH. Este é um aspecto interessante, pois torna a rede



mais robusta, uma vez que mitiga os efeitos provenientes da perda de dados, tal qual ocorre em um sistema de transmissão de vídeo em tempo real.

Com base nos resultados das simulações, algumas diretrizes de projeto poderiam ser tomadas no estabelecimento de uma rede PM-AH, de modo a não comprometer o desempenho no recebimento das mensagens de dados no Provedor de Serviços.

Dentre essas diretrizes de projeto, o mais importante é o correto dimensionamento da rede. No caso específico do PS, isso remete a escolha do *switch* para realizar a conexão entre os dispositivos biomédicos. Deve-se ter certeza que o *switch* escolhido suporta o tráfego gerado pelos nós da rede no cenário de pior caso, ou seja, o tamanho do *buffer* presente na porta onde o PS estará conectado deve ser determinado pela seguinte equação:

$$T_{buffer} = N_{leitos} * N \quad (\text{Equação 4.1})$$

Em que:

$T_{buffer}$ : Tamanho mínimo do *buffer* (em *bytes*) presente nas portas do *switch*.

$N_{leitos}$ : Quantidade de leitos presentes no ambiente hospitalar.

$N$ : Tamanho das mensagens trocadas na rede (em *bytes*).

Outras diretrizes de projeto podem ser voltadas à busca de melhoria do atraso médio na transmissão de dados dos dispositivos biomédicos para o Provedor de Serviços. Nesse caso pode-se realizar uma divisão da rede em outras menores as quais, conforme visto nos resultados das simulações, acarretam um atraso de comunicação menor.

Como exemplo dessa diretriz poder-se-ia dividir a rede hospitalar correspondente a simulação com 60 leitos com 6 dispositivos cada em 6 redes menores cada uma contendo 10 leitos cada. Dessa forma, o tempo de transmissão de mensagens de dados seria bem menor quando comparada à rede original.

Todas essas questões de projeto devem levar em consideração as necessidades do ambiente hospitalar e as aplicações que irão interagir com os elementos deste. Dessa forma, todos os pontos devem ser levados em conta de modo a construir uma estrutura que permita a transferência de mensagens de dados visando extrair o melhor desempenho possível.

#### 4.2.3. Provedor de Serviços: Uma nova abordagem

Os experimentos demonstraram que o PS, quando submetido a uma carga intensa de dados, perde a estabilidade quanto ao determinismo e, nos casos em que o fluxo de mensagens é maior que a capacidade de processamento do *switch*, o PS deixa de receber todas as

mensagens de dados, visto que algumas são descartadas na porta do *switch* em que o PS está conectado.

Neste contexto, é importante mencionar que os efeitos ocasionados não geram impactos que depreciem o funcionamento do PM-AH, por dois motivos especificamente:

1. O não determinismo ocorrido no PS não afeta a rede de monitoramento de pacientes regida pelo PM-HA, visto que os ciclos temporais continuam executando livre de contenção, enfileiramentos e com baixa variação no atraso (Valentim, 2008);
2. A perda de dados ocasionada pelo descarte de pacotes não é desejável, mas, como os processos de monitoramento de pacientes são lentos, o PM-AH passa a gerar dados redundantes (o nível de amostragem do PM-AH funciona na casa de microssegundos enquanto a planta opera na casa de segundos), de modo que, o sistema de armazenamento de dados no PS só será prejudicado em casos extremos.

Observando os itens mencionados é possível inferir que o problema do determinismo na rede de controle (monitoramento de pacientes) está resolvido, todavia, o PS, mesmo com os níveis de redundância gerados pelas mensagens de dados do PM-AH, é um ponto de gargalo que necessita de maior robustez dada à importância do Provedor de Serviços para rede PM-AH.

Norteados por este princípio, o presente trabalho aborda uma alternativa para a execução do monitoramento de pacientes através do uso de políticas de controle na entrega de mensagens de dados ao Provedor de Serviços, com vista, a garantir a escalabilidade e a robustez do PS.

Nessa abordagem, os dispositivos biomédicos não enviam mais mensagens de dados diretamente ao PS. Ao invés disso, os mestres de cada leito armazenam em *buffer* o conteúdo de todas as mensagens trocadas dentro do leito durante certo número de ciclos temporais (Valentim, 2008).

Esse volume de dados estará presente no mestre até o momento em que esse receber uma mensagem de *token* originada pelo Provedor de Serviços. Após o recebimento dessa mensagem, o mestre envia todo o volume de dados ao Provedor de Serviços que cadastra as informações no sistema (Souza, 2008).

A implementação desse mecanismo pode ser feita pelo envio de mensagens de *token* aos dispositivos mestres em períodos de tempo determinados pela prioridade de cada leito, ou simplesmente por um sistema de *polling* baseado em fila.

Para efeitos de comparação entre as abordagens, foram realizadas quatro simulações da rede PM-AH adotando a nova abordagem para o Provedor de Serviços. As simulações executadas consideraram os mesmos cenários de teste das simulações anteriores, com as mesmas especificações, preocupando-se também em considerar os mesmo parâmetros de desempenho.

Nas simulações utilizando a nova abordagem adotou-se que o mestre de cada leito armazenava um volume de dados correspondente a troca de mensagens de um ciclo temporal realizado no seu leito hospitalar, sendo estas enviadas no instante em que este mestre recebe um *token* enviado pelo PS. Nessas simulações, a transmissão escolhida para ser monitorada foi a do mestre do leito 1 ao Provedor de Serviços.

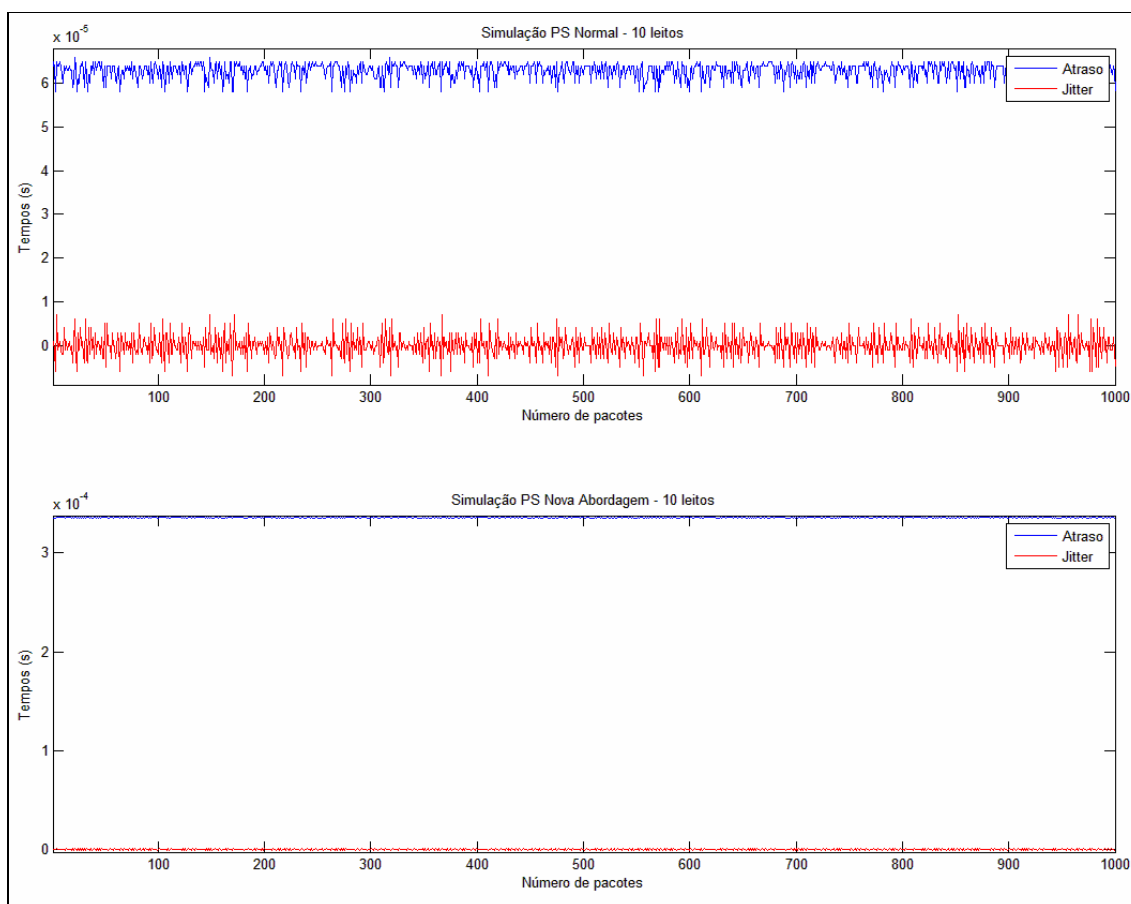
A primeira simulação executada foi do cenário de teste que considerava 10 leitos com 6 dispositivos cada.

A Tabela 4.10 descreve as métricas de desempenho obtidas nessa simulação e faz um comparativo dos valores dessas métricas com os obtidos na simulação que considerava o mesmo cenário de testes, mas com recebimento de mensagens de dados no PS sem o uso de passagem de *token*.

A Figura 4.10 ilustra o atraso e *jitter* para a transmissão de dados usando a abordagem do PS com passagem de *token* e faz um comparativo desses com o atraso e *jitter* obtidos na simulação do mesmo cenário de teste, mas com recebimento de mensagens de dados no PS sem o uso de passagem de *token*.

**Tabela 4.10 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 10 leitos com 6 dispositivos cada).**

<b>Parâmetros de desempenho</b>	<b>Valor (PS sem passagem de <i>token</i>)</b>	<b>Valor (PS com passagem de <i>token</i>)</b>
Atraso médio	$6.2918 \times 10^{-5}$ s	$3.3563 \times 10^{-4}$ s
<i>Jitter</i> Médio	$6.0000 \times 10^{-9}$ s	$2.7736 \times 10^{-16}$ s
Desvio padrão	$1.9102 \times 10^{-6}$ s	$4.8250 \times 10^{-7}$ s
Número de pacotes perdidos (transmissão)	0 pacotes	0 pacotes
Número de pacote perdidos (rede)	0 pacotes	0 pacotes
Número de alarmes gerados	683 pacotes	595 pacotes
Número de pacotes gerados	120.683 pacotes	20.609 pacotes



**Figura 4.10 - Comparativo do atraso e *jitter* na transmissão de dados entra a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 10 leitos com 6 dispositivos cada).**

Analisando os resultados da simulação observa-se que tanto a abordagem do PS sem passagem de *token* como a com passagem de *token* não apresentam perda de mensagens de dados, ou seja, em ambos os casos o Provedor de Serviços irá cadastrar todas as informações referentes às medidas dos sinais vitais dos pacientes. Todavia, um aspecto que merece destaque é o fato da transmissão das mensagens de dados na simulação que faz uso do PS com passagem de *token* ser determinística, ou seja, a entrega dos dados é realizada em instantes de tempo previsíveis o que facilita o desenvolvimento de aplicações de monitoramento de pacientes em tempo real. Esse fato pode ser verificado através do baixo *jitter* ao longo da transmissão dos pacotes de dados.

Com relação ao atraso médio das transmissões nestes experimentos, foi verificado que, o tempo médio na abordagem em que o PS faz uso do sistema de passagem de *token* ( $335.63\mu\text{s}$ ) é cinco vezes maior quando comparado com a abordagem na qual o PS não faz

uso do sistema de passagem de *token* (62.9 $\mu$ s). A justificativa para esse fato, decorre do tamanho das mensagens enviadas ao Provedor de Serviços, visto que, quando o PS utiliza o mecanismo de passagem de *token*, o dispositivo mestre de um dado leito passa a armazenar todas as mensagens de dados trocadas dentro de um ciclo temporal (Souza, 2008). Na abordagem sem passagem de *token* cada dispositivo envia a mensagem diretamente ao PS (Valentim, 2008). Nesse aspecto, o tamanho da mensagem a ser enviada ao PS é bem maior na abordagem que faz uso de passagem de *token* o que impacta substancialmente no atraso médio das transmissões.

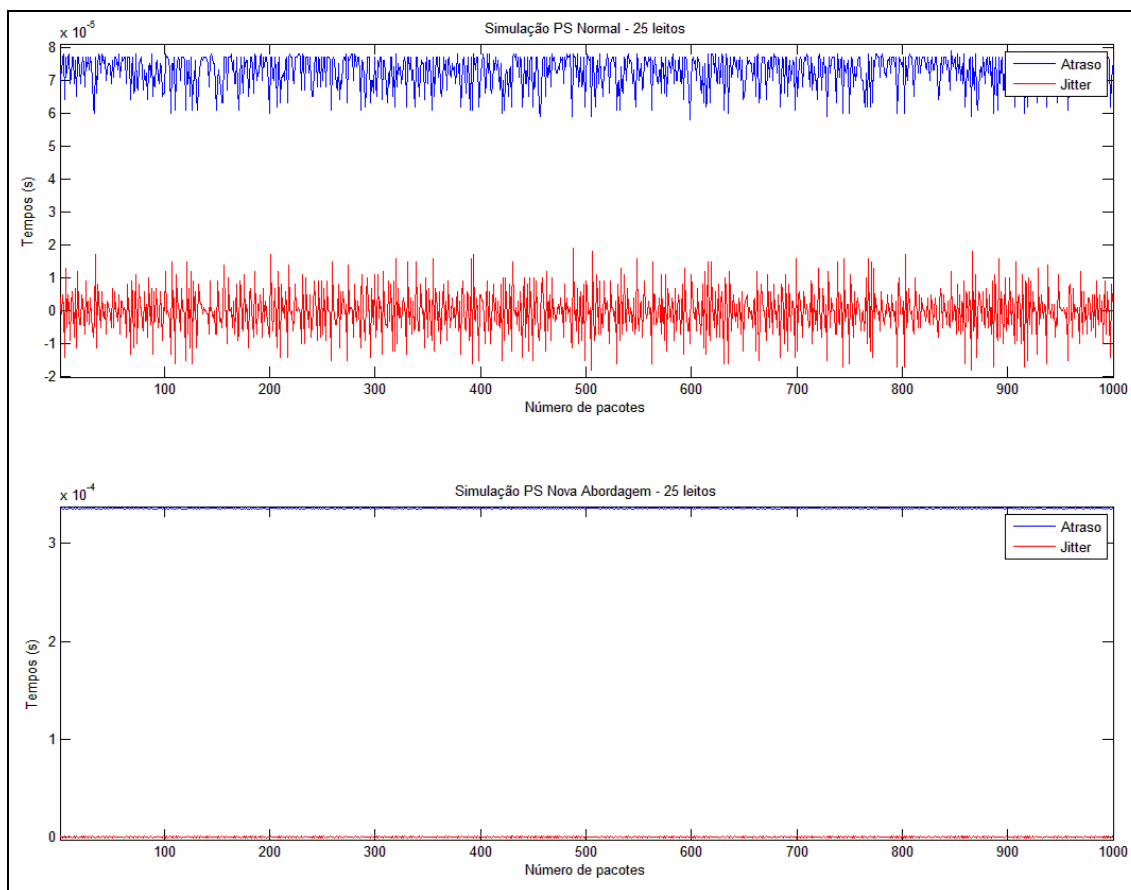
A segunda simulação executada foi do cenário de 25 leitos com 6 dispositivos cada. Essa simulação é importante, pois permite avaliar o desempenho na entrega de mensagens de dados ao Provedor de Serviços quando a rede é submetida a diferentes níveis de tráfego, uma vez que, o aumento no número de leitos implica no aumento do número de dispositivos na rede, fator que acarreta um nível de tráfego mais elevado (Valentim, 2008).

A Tabela 4.11 descreve as métricas de desempenho obtidas na simulação e faz um comparativo dessas métricas com as obtidas na simulação do mesmo cenário de teste que considerava o recebimento de mensagens de dados no PS sem o mecanismo de passagem de *token*.

A Figura 4.11 ilustra o atraso e o *jitter* para a transmissão de dados usando a abordagem do PS com passagem de *token* e faz um comparativo desses com o atraso e *jitter* obtidos na simulação do mesmo cenário de teste, mas considerando que o Provedor de Serviços não fazia uso do mecanismo de passagem de *token*.

**Tabela 4.11 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 25 leitos com 6 dispositivos cada).**

<b>Parâmetros de desempenho</b>	<b>Valor (PS sem passagem de <i>token</i>)</b>	<b>Valor (PS com passagem de <i>token</i>)</b>
Atraso médio	7.2671x10 <sup>-5</sup> s	3.3567x10 <sup>-4</sup> s
<i>Jitter</i> Médio	7.0000x10 <sup>-9</sup> s	1.0000x10 <sup>-9</sup> s
Desvio padrão	4.9664x10 <sup>-6</sup> s	4.6972x10 <sup>-7</sup> s
Número de pacotes perdidos (transmissão)	0 pacotes	0 pacotes
Número de pacote perdidos (rede)	0 pacotes	0 pacotes
Número de alarmes gerados	1.655 pacotes	1.580 pacotes
Número de pacotes gerados	301.655 pacotes	51.619 pacotes



**Figura 4.11 - Comparativo do atraso e *jitter* na transmissão de dados entra a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 25 leitos com 6 dispositivos cada).**

Nessas simulações executadas, ambas não apresentam perda de mensagens de dados. Assim como no cenário de testes anterior, a simulação do Provedor de Serviços sob a abordagem que faz uso de passagem de *token* apresentou determinismo na transmissão das mensagens de dados enquanto a outra abordagem mostrou-se não-determinística.

Um aspecto interessante nessa nova simulação foi que, mesmo com o aumento no número de dispositivos na rede (ocasionado pelo aumento no número de leitos), a abordagem do PS com passagem de *token* não gerou impacto no atraso médio da transmissão das mensagens de dados ao Provedor de Serviços. Dessa forma, essa nova abordagem mantém o comportamento da rede constante independentemente da quantidade de nós presentes nela.

Essa característica é interessante, pois assegura o funcionamento adequado na recepção de dados no Provedor de Serviços independente do tamanho do *buffer* nas portas do *switch* (Equação 4.1).

A terceira simulação executada foi a do cenário de teste com 50 leitos com 6

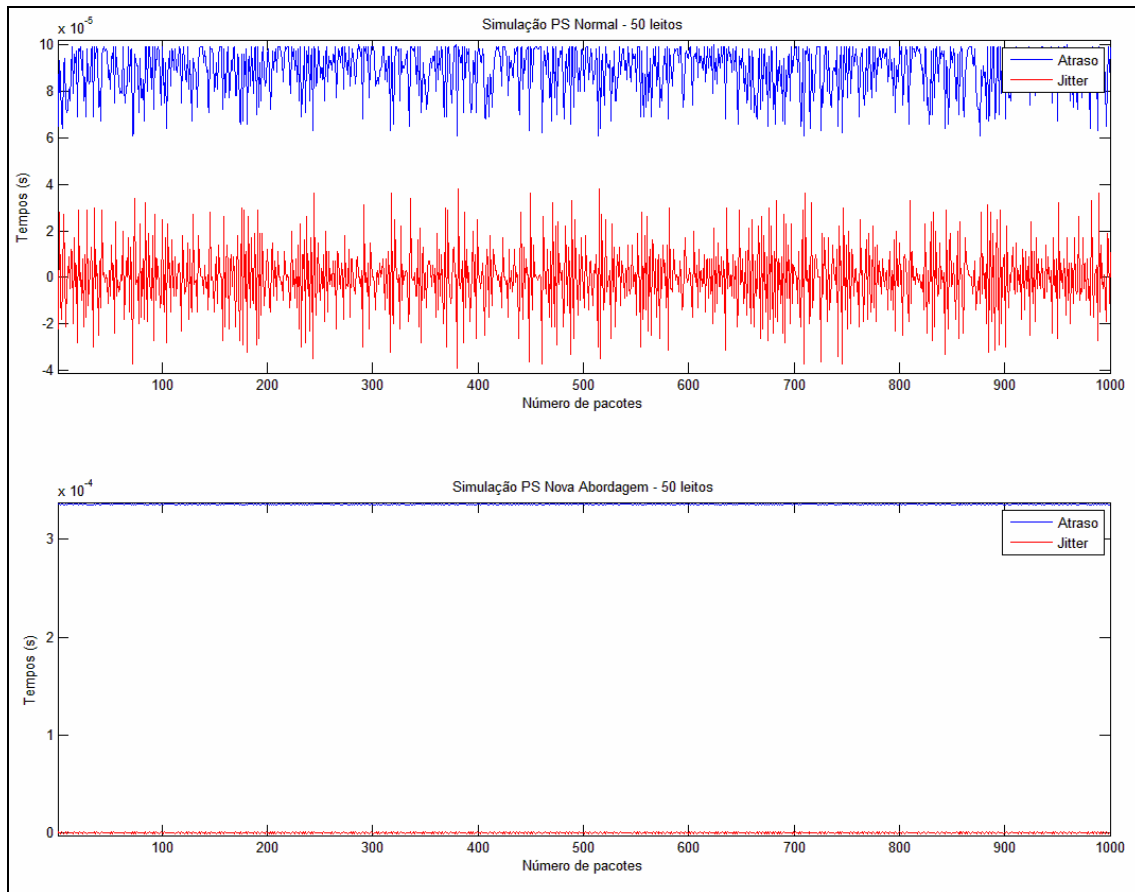
dispositivos cada. Essa simulação é importante, pois serve para avaliar o desempenho na entrega de mensagens de dados ao PS quando a rede é submetida à quantidade máxima de tráfego suportada pela mesma (cenário do pior caso). Esse fato é justificado em razão do tamanho do *buffer* adotado para as portas do *switch* na simulação foi de 50 mensagens. Como os leitos operam em paralelo, então, em um dado instante um dispositivo de cada leito pode enviar seus dados totalizando assim 50 mensagens a serem entregues ao Provedor de Serviços. A diferença na simulação do PS utilizando passagem de *token* é que essas 50 mensagens não serão redirecionadas ao Provedor de Serviço de uma só vez, como ocorre no caso em que o PS não utiliza mensagens de *token*. Ao contrário, o mestre de cada leito armazenará as mensagens enviadas pelos dispositivos de seus respectivos leitos em um *buffer* para redirecionar ao PS logo após receber uma mensagem de *token* originada dele.

A Tabela 4.12 descreve as métricas de desempenho obtidas na simulação e faz um comparativo dessas métricas com as obtidas na simulação do mesmo cenário de teste que considerava o recebimento de mensagens de dados no PS sem o uso de passagem de *token*.

A Figura 4.12 ilustra o atraso e *jitter* para a transmissão de dados usando a abordagem do PS com passagem de *token* e faz um comparativo desses com o atraso e *jitter* obtidos na simulação do mesmo cenário de teste, mas considerando que o Provedor de Serviços não fazia uso da passagem de *token*.

**Tabela 4.12 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 50 leitos com 6 dispositivos cada).**

<b>Parâmetros de desempenho</b>	<b>Valor (PS sem passagem de <i>token</i>)</b>	<b>Valor (PS com passagem de <i>token</i>)</b>
Atraso médio	$8.893 \times 10^{-5} \text{ s}$	$3.3565 \times 10^{-4} \text{ s}$
<i>Jitter</i> Médio	$1.2000 \times 10^{-8} \text{ s}$	$1.0000 \times 10^{-9} \text{ s}$
Desvio padrão	$1.0183 \times 10^{-5} \text{ s}$	$4.7625 \times 10^{-7} \text{ s}$
Número de pacotes perdidos (transmissão)	0 pacotes	0 pacotes
Número de pacote perdidos (rede)	0 pacotes	0 pacotes
Número de alarmes gerados	3.341 pacotes	3.124 pacotes
Número de pacotes gerados	603.341 pacotes	103.198 pacotes



**Figura 4.12 - Comparativo do atraso e *jitter* na transmissão de dados entra a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 50 leitos com 6 dispositivos cada).**

A análise comparativa dos resultados obtidos nas simulações que fazem uso do PS com passagem de *token* e sem passagem de *token* para o cenário de teste composto por 50 leitos de 6 dispositivos cada, demonstram que, em ambos os casos, não houve perda de mensagens de dados.

Assim como nos cenários anteriores, o atraso médio na transferência de mensagens de dados ao Provedor de Serviços manteve-se igual para a abordagem que utiliza o PS com passagem de *token*. Esse fato comprova a eficiência dessa abordagem quando a rede hospitalar é submetida a diferentes níveis de tráfego, além de atestar o bom funcionamento na entrega das mensagens de dados, mesmo quando a rede é submetida a sua carga máxima.

A quarta simulação executada foi do cenário de teste com 60 leitos de 6 dispositivos cada. Essa simulação é importante, pois serve para avaliar o desempenho na entrega de mensagens de dados ao Provedor de Serviços quando a rede é submetida a um nível de



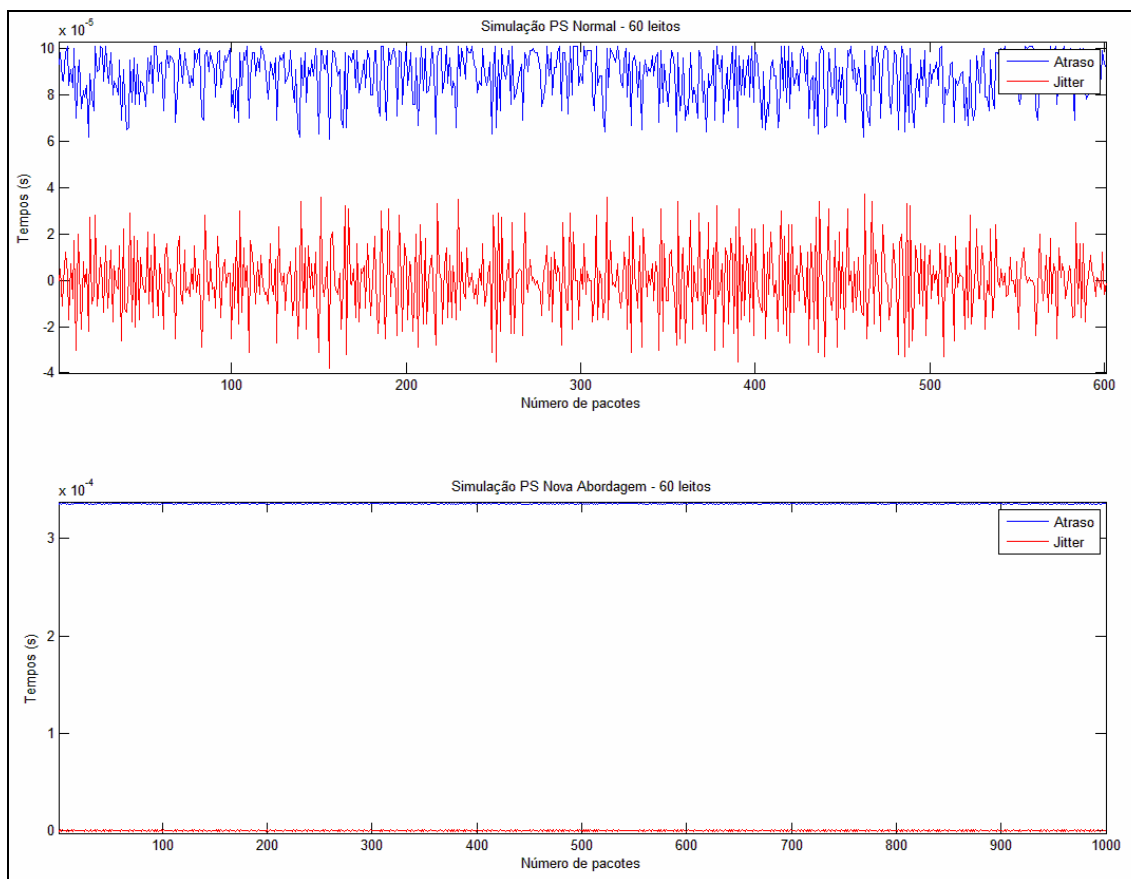
tráfego maior que o suportado pela mesma. A justificativa para esse fato é a mesma do cenário de testes com 50 leitos, ou seja, como os leitos operam em paralelo em um dado momento cada dispositivo de cada leito pode enviar seus dados totalizando assim 60 mensagens a serem entregues ao Provedor de Serviços. Esse número, porém é maior que o tamanho do *buffer* adotado na simulação para as portas do *switch* (50 mensagens).

A Tabela 4.13 descreve as métricas de desempenho obtidas na simulação e faz um comparativo dessas métricas com as obtidas na simulação do mesmo cenário de teste considerando o PS sem o uso de passagem de *token*.

A Figura 4.13 ilustra o atraso e *jitter* para a transmissão de dados usando as duas abordagens (Provedor de Serviços com e sem passagem de *token*).

**Tabela 4.13 - Comparativo das métricas de desempenho levantadas para a transmissão de dados entre a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 60 leitos com 6 dispositivos cada).**

<b>Parâmetros de desempenho</b>	<b>Valor (PS sem passagem de <i>token</i>)</b>	<b>Valor (PS com passagem de <i>token</i>)</b>
Atraso médio	$8.8115 \times 10^{-5} \text{ s}$	$3.3566 \times 10^{-4} \text{ s}$
<i>Jitter</i> Médio	$1.6639 \times 10^{-9} \text{ s}$	$6.1176 \times 10^{-17} \text{ s}$
Desvio padrão	$1.0769 \times 10^{-5} \text{ s}$	$4.7395 \times 10^{-7} \text{ s}$
Número de pacotes perdidos (transmissão)	399 pacotes	0 pacotes
Número de pacote perdidos (rede)	53.983 pacotes	0 pacotes
Número de alarmes gerados	3.984 pacotes	3.785 pacotes
Número de pacotes gerados	723.984 pacotes	123.887 pacotes



**Figura 4.13 - Comparativo do atraso e jitter na transmissão de dados entra a abordagem do PS sem passagem de *token* e com passagem de *token* (simulação para 60 leitos de 6 dispositivos cada).**

A análise dos resultados dessa simulação remete pontos interessantes para a abordagem do PS utilizando mensagens de *tokens*. A manutenção do atraso médio e determinismo já foram verificados em cenários de teste anteriores. Nessa simulação verifica-se que a abordagem do PS utilizando mensagens de *token* não tem perda de mensagens, algo que não é verificado na simulação do mesmo cenário de testes utilizando a abordagem do PS sem mensagens de *token*.

Esse aspecto é justificado pelo fato que, no cenário em que o PS não faz uso de mensagens de *token*, as mensagens de dados são enviadas diretamente ao Provedor de Serviços, dessa forma, como os leitos operam em paralelo em um dado instante um dispositivo de cada leito envia suas mensagens de dados as quais devem ser entregues ao PS. Como o cenário de teste considera 60 leitos, então existirão 60 mensagens de dados a serem redirecionadas a porta do *switch* no qual o PS está conectado, em virtude do tamanho do *buffer* das portas do *switch* adotado nas simulações ser de 50 mensagens então alguns dos

pacotes de dados a serem entregues ao PS serão perdidos em virtude do descarte ocasionado pela sobrecarga da capacidade máxima de armazenamento (50 mensagens) presente na porta do *switch* em que o Provedor de Serviços está conectado.

Na simulação utilizando a abordagem do PS com passagem de *token* o dispositivo biomédico não envia mais as mensagens de dados diretamente ao Provedor de Serviços, ao invés disso, quando essas mensagens chegam ao mestre do leito (proveniente do envio ao grupo *multicast*, segundo o esquema de *publisher-subscribe*, representativo do leito nos quais os dispositivos biomédicos estão associados), esse armazena o conteúdo dessas mensagens em um *buffer* interno (Souza, 2008). Esse conteúdo permanecerá no *buffer* do mestre até o momento em que esse recebe uma mensagem de *token* originária do Provedor de Serviços, nesse instante, o mestre envia toda a massa de dados armazenada em seu *buffer* ao PS (Valentim, 2008).

Como o Provedor de Serviços regula os instantes em que serão recebidas mensagens de dados não ocorre enfileiramentos na porta do *switch* em que esse está conectado, com isso, não ocorre descarte de mensagens (Souza, 2008).

Essa abordagem do PS utilizando passagem de *token* é interessante, pois permite o desenvolvimento de aplicações de monitoramento de pacientes em tempo real, em decorrência do determinismo na entrega de mensagens, e garante a robustez do sistema, uma vez que, todos os dados enviados pelos dispositivos biomédicos são cadastrados.

Embora tenha uma série de vantagens, a abordagem do PS com passagem de mensagens de *token* apresenta alguns aspectos negativos também. Um aspecto significativo dessa abordagem, diz respeito ao armazenamento de dados no mestre de cada leito, esse fator certamente demandará mais recurso computacional.

Outro ponto que merece destaque no uso do Provedor de Serviços com passagem de *token* é a necessidade de toda reestruturação do PM-AH. As mensagens de dados são mensagens enviadas a grupos *multicast* correspondentes aos leitos hospitalares em que os dispositivos biomédicos estão inscritos (Valentim, 2008). Na abordagem do Provedor de Serviços com passagem de *token* as mensagens são enviadas diretamente aos mestres, ao invés do PS. Como essas mensagens são enviadas a grupos *multicast* correspondentes aos leitos hospitalares, isso implica que o Provedor de Serviços não pode estar inscrito nesses leitos, desta forma, implicando alteração no protocolo PM-AH.

Desse modo, para viabilizar tal proposta, deve ser feito um estudo de como reestruturar o PM-AH para que possa ser incorporado o Provedor de Serviços com passagem de *token* e deve ser levantado até que ponto a adição da funcionalidade de armazenamento de

dados no mestre de cada leito pode influenciar no desempenho do protocolo e também deste dispositivo.

Realizado esse estudo, pode ser viável a adoção do Provedor de Serviços com passagem de *token* em virtude da série de benefícios agregados, porém todos os pontos mencionados devem ser avaliados de forma criteriosa, de modo que sejam cumpridos os requisitos temporais impostas pela rede de automação hospitalar.

## 5. CONCLUSÃO

O PM-AH é uma alternativa interessante para o estabelecimento da rede de comunicação entre dispositivos biomédicos por ser um protocolo que atende aos requisitos emergentes da automação hospitalar e por utilizar o padrão de comunicação *Ethernet*, proporcionando assim, uma redução de custos de instalação em virtude do uso da infraestrutura existente no ambiente hospitalar (Souza, 2008).

Um dos elementos mais importantes da rede PM-AH é o Provedor de Serviços, pois é responsável pelo estabelecimento e manutenção da comunicação entre os dispositivos biomédicos (Valentim, 2008). Em virtude da sua importância, é interessante que haja um estudo sobre quais funcionalidades o Provedor de Serviços deve dispor e como ele irá executá-las, para tanto, deve-se construir modelos que estabeleçam diretrizes de projetos para sua implementação.

Nesse aspecto, o presente trabalho apresentou contribuições relevantes na medida em que, especificou e verificou formalmente um modelo para o banco de dados para ser utilizado pelo Provedor de Serviços. Neste mesmo íterim, também especificou, validou e verificou formalmente um modelo em Rede de Petri que garante uma implementação do PS livre de *deadlock*.

Além dessas contribuições, o presente trabalho abordou também a realização de análises de desempenho sobre as tarefas executadas pelo Provedor de Serviços, sobre duas perspectivas: rede e custo de processamento derivado das consultas do banco de dados do PS. Essas análises são importantes, pois permitem uma avaliação da estrutura proposta pelo PM-AH, de modo a verificar até onde o desempenho da rede hospitalar é satisfatório.

Outra contribuição do trabalho foi à descrição de uma nova abordagem para o PS, de modo, a obter um melhor desempenho no recebimento das mensagens de dados oriundas dos dispositivos biomédicos. Nesse aspecto, foram realizadas análises de desempenho sobre essa nova abordagem e foi avaliado qual o ganho da mesma em comparação à abordagem tradicional do PM-AH. Além da avaliação do ganho obtido, o presente trabalho realiza considerações sobre o impacto dessa nova abordagem no protocolo, destacando pontos que deveriam ser alterados no mesmo, e quais as funcionalidades que deveriam ser adicionadas nos dispositivos biomédicos.

Com relação aos trabalhos em perspectiva abordados, tem-se a construção de sistemas supervisórios, implementação dos modelos em Rede de Petri descritos e reestruturação do PM-AH adicionando funcionalidades de qualidade de serviço (QoS) na transferência de mensagens de dados.

## **5.1. Trabalhos em perspectiva**

### **5.1.1. Sistema supervisório**

Esse trabalho aborda a construção de sistemas supervisórios que atuem em conjunto com o Provedor de Serviços de modo a permitir o acesso a informações referentes aos pacientes dos leitos da rede hospitalar.

Para a realização desse trabalho, haveria a necessidade de um estudo relacionado às quais funcionalidades que o sistema deveria dispor e como os usuários poderiam interagir com o mesmo de modo a obter as informações desejadas.

### **5.1.2. Implementação dos modelos em Rede de Petri**

O presente trabalho realiza a implementação de uma parte da Rede de Petri descrita na Figura 3.9 (mais especificamente a parte referente ao processamento de mensagens de requisição de inscrição de dispositivos).

Trabalhos em perspectiva poderiam abordar a implementação das demais partes da Rede de Petri em questão podendo, eventualmente, descrever a realização de análises de desempenho sobre essas implementações similar a análise de desempenho efetuada no presente trabalho.

### **5.1.3. Reestruturação do PM-AH adicionando funcionalidades de Qualidade de Serviço (QoS)**

Conforme visto, a nova abordagem do Provedor de Serviços para o recebimento de mensagens de dados exige a reestruturação do PM-AH. Sendo assim, um trabalho em perspectiva seria a descrição de como seria feita essa reestruturação abordando os pontos que seriam modificados no protocolo e como seriam feitas essas modificações.

Nesse aspecto, outro trabalho interessante seria a adoção de funcionalidades destinadas ao provimento de Qualidade de Serviço (QoS) na transferência de mensagens de dados. Conforme visto, na nova abordagem para o Provedor de Serviços, a transferência dessas mensagens é feita por meio de passagem de *token* do PS para os mestres dos leitos, em

que, ao receberem essas mensagens, tais mestres transferiam toda a massa de dados armazenada em um *buffer* interno referente à comunicação de seus leitos durante um número de ciclos temporais arbitrário. Dessa forma poder-se-iam implantar políticas de qualidade de serviços (QoS) no processo de aquisição de dados dos dispositivos biomédicos através da atribuição prioridades aos leitos, de modo que, ocorra um envio de mensagens de *token* para os mestres dos leitos que possuem maior prioridade.

Desse modo, haveria a necessidade também de especificar quais os parâmetros que seriam utilizados na atribuição de prioridades aos leitos, sendo assim, a definição desses parâmetros seria outro ponto a ser tratado no trabalho em perspectiva.

## REFERÊNCIAS

- Dietrich, D., Sauter, T. Evolution Potentials for Fieldbus Systems. WFCS 2000, IEEE Workshop on Factory Communication Systems. Porto, Portugal, September 2000.
- Nitzan, D. and Rosen, C. A. Programmable Industrial Automation. Transactions on Computers. Volume C-25, Issue 12, Dec. 1976 Page(s):1259 – 1270.
- Pedreiras, P., Almeida, L., Gai, P. and Giorgio, B. “FTT-Ethernet: A Flexible Real-Time communication Protocol That Supports Dynamic QoS Management on Ethernet-Based Systems”. IEEE Transactions on Industrial Informatics, Vol. 1, N°. 3, August 2005.
- Protocolo Multiciclos para Automação Hospitalar, Ricardo Alexsandro de Medeiros Valentin - Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica e Computação (PPGEEC), Universidade Federal do Rio Grande Do Norte (UFRN), 2008.
- Uma Análise de desempenho sobre Redes Ethernet-PM-AH (Protocolo Multiciclos Para Automação Hospitalar) Via Simulação, Vinícius Samuel Valério de Souza – Trabalho de Conclusão de Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte (UFRN), 2008.
- Brooks, J.; Brooks, L. Automation in the medical field. Engineering in Medicine and Biology Magazine, IEEE Volume 17, Issue 4, July-Aug. 1998 Page(s): 76, 1.
- Valentim, R. A. M. ; Morais, A. H. ; Souza, V. S. V. ; Araújo J, H. B ; Brandão, G. B. ; Guerreiro, A. M. G. . Rede de Controle em Automação Hospitalar: Um Protocolo Multiciclos para Automação Hospitalar sobre IEEE 802.3 com IGMP Snooping. Revista Ciência e Tecnologia, v. 11, p. 19, 2009.
- Florentino, G. H. P; Bezerra, H. U; Araújo Júnior, H. B; Araújo, M. X; Valentim, R. A. M; Morais, A. H. F; Guerreiro A. M. G, Brandão, G. B.; Araújo, C. A. P. Hospital Automation RFID-Based: Technology Stored In Smart Cards. World Congress on



Engineering (WCE 2008), ICSBB 2008 - The 2008 International Conference of Systems Biology and Bioengineering. London, U.K., 2-4 July 2008.

Valentim, R. A. de M; Morais, A. H. F, Bezerra, H. U; Araújo Júnior, H. B; Xavier, M. A; Guerreiro; A. M. G; Brandão, G. B; Araújo, C. A. P. MP-HA: Multicycles Protocol for Hospital Automation over Multicast. World Congress on Engineering (WCE 2008), ICSBB 2008 - The 2008 International Conference of Systems Biology and Bioengineering. London, U.K., 2-4 July 2008.

Valentim, R. A. de M; Morais, A. H. F, Bezerra, H. U; Araújo Júnior, H. B; Xavier, M. A; Guerreiro; A. M. G; Brandão, G. B; Araújo, C. A. P. MP-HA: Multicycles Protocol for Hospital Automation over multicast with IEEE 802.3. Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference. Page(s): 979 – 984. Daejeon, South Korea, 2008.

Araújo J, H. B ; Bezerra, H. U ; Xavier, M. A. ; Souza, V. S. V. ; Valentim, R. A. M. ; Morais, A. H. ; Guerreiro, A. M. G. ; Brandão, G. B. RFID Aplicado a Automação Hospitalar: Desenvolvimento de um Sistema para Automação de Laboratórios de Análise Clínica. In: XVII Congresso Brasileiro de Automática, 2008, Juiz de Fora. XVII Congresso Brasileiro de Automática, 2008.

Florentino, G. H. P. ; Paz, C. A. P ; Bezerra, H. U ; Araújo J, H. B ; Xavier, M. A. ; Souza, V. S. V. ; Valentim, R. A. M. ; Morais, A. H. ; Guerreiro, A. M. G. ; Brandão, G. B. . Hospital Automation System RFID-Based: Technology Embedded In Smart Devices (Cards, Tags and Bracelets). In: Engineering in Medicine and Biology Society, 2008, Vancouver. Engineering in Medicine and Biology Society, 2008.

Valentim, R. A. M. ; Morais, A. H. ; Carvalho, D. R. ; Brandão, G. B. ; Araújo J, H. B ; Bezerra, H. U ; Souza, V. S. V. ; Guerreiro, A. M. G. ; Paz, C. A. P. Multicycles Protocol for Hospital Automation Based on Multicast Networking. In: International Conference on Control, Automation and Systems, 2008, Seoul. International Conference on Control, Automation and Systems, 2008.

Valentim, R. A. M. ; Morais, A. H. ; Araújo J, H. B ; Xavier, M. A. ; Bezerra, H. U ; Souza, V. S. V. ; Amorin, L. G. P ; Guerreiro, A. M. G. ; Brandão, G. B. ; Paz, C. A. P . Medium Access Control: Multicycles Protocol for Hospital Automation over Multicast with IEEE 802.3. In: 34th Conference of IEEE Industrial Electronics Society, 2008, Orlando. 34th Conference of IEEE Industrial Electronics Society, 2008.

Valentim, R. A. M. ; Morais, A. H. ; Araújo J, H. B ; Bezerra, H. U ; Souza, V. S. V. ; Guerreiro, A. M. G. ; Brandão, G. B. ; Paz, C. A. P . Hospital Network: Multicycles Protocol for Hospital Automation over IEEE 802.3 with IGMP Snooping. In: International Conference on Industrial Technology, 2008, Churchill. International Conference on Industrial Technology, 2009.

Brito, A. E. M., Brasileiro, F. V., Leite C. E., Buriti, A. C. Comunicação Ethernet em Tempo-Real para uma Rede de Microcontroladores, Anais do XV Congresso Brasileiro de Automática (CBA 2004) – Brasil, setembro 2004.

DATE, C.J. Introdução a Sistemas de Banco de Dados. Editora Campus. 8ª Edição

Dolejs, O., Smolik, P., e Hanzalek Z. “On the Ethernet use for real-time publishsubscribe based applications”. In 5th IEEE International Workshop on Factory Communication Systems, Vienna, Austria, Sep. 2004.

Carreiro, F., Moraes, R., Fonseca, J. A e Vasques, F. "Real-Time Communication in Unconstrained Shared Ethernet Networks: The Virtual Token-Passing Approach" submitted at Emerging Technologies and Factory Automation – ETFA, Catania Italy, 2005.

IEEE 802.3/ISO 8802-3 - Information processing systems – Local area networks - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 2nd edition, 21 September 1990.

Moraes, R.; Vasques, F., "Real-time traffic separation in shared Ethernet networks: simulation analysis of the h-BEB collision resolution algorithm," Embedded and Real-Time

- Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on, vol., no. pp. 89- 92, 17-19 Aug. 2005.
- RFC 4541. Christensen, M.; Thrane & Thrane; Kimball, K. IGMP and MLD Snooping Switches Considerations. Hewlett-Packard, May, 2006.
- Projeto de Banco de Dados, Carlos Alberto Heuser, Editora: Digital Source, 4ª Edição, Ano: 1998.
- Cood, E. F. A relational Model of Data for Large Shared Data Banks. Communication of the ACM, 13 (06) June 1970.
- Viégas, R.; Valentim, R.A.M.; Texeira, D.G.; Guedes, L.A. Performance Measurements of Protocols to Ethernet Real-Time Applications. Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on 20-22 Sept. 2006 Page(s):1352 – 1355.
- Jain, R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley-Interscience, New York, NY, April 1991.
- Coutinho, M. M. Network Simulator. Guia Básico para Iniciantes. Apostila. Engenharia Elétrica – UFPA. Disponível em: <http://www.di.ufpe.br/~sfd/universo/sim/nsr1.pdf>. 2003.
- PETRI, C.A. Kommunikation mit Automaten. Schriften des IIM Nr. 2, Institut für Instrumentelle Mathematik, Bonn, 1962. English Translation: Technical Report RADC-TR-65-377, Griffiths Air Force Base, New York, v. 1, Suppl.1, 1966.
- Maciel, P.R.M., Lins, R.D., Cunha, P.R.F., Introdução às Redes de Petri e Aplicações, 10ª Escola de Computação, Campinas, Julho 1996.
- Pere Bonet, Catalina M. Llado, Ramon Puigjaner, PIPE2 v 2.5 : a Petri Net Tool for Performance Modeling, Departament de Ciències Matemàtiques i Informàtica, Universitat de les Illes Balears, Palma de Mallorca, Spain, 2007.

J. Postel, "User Datagram Protocol," RFC 768, Aug. 1980.

J. Postel, "Internet Protocol: DARPA Internet Program Protocol Specification," RFC 791, Sept. 1981.

Tanenbaum, A. S. "Computer Networks". 4rd. Ed., Prentice- Hall, 2003.

Tanenbaum, A. S. "Sistemas Operacionais Projeto e Aplicações". 2ª Edição. Ed. Bookman, 2000.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)